HELSINGIN KAUPPAKORKEAKOULU
**Johtamisen laitos**

SALMINEN
4.12.99
A517

SOFTWARE PROCESS IMPROVEMENT:

A CASE STUDY OF USING THE CAPABILITY MATURITY MODEL (CMM)



Organisaatiot ja johtaminen,
pro gradu-tutkielma
Kaisa Uronen, 22167-0
syyslukukausi 1997

__________Johtamisen__________ laitoksen

johtajan päätöksellä 4 / 12 1997 hyväksytty

arvosanalla MAGNA CUM LAUDE APPROBATUR

TARKASTAJAT: ANTTI AINAMO JA VIRPI TUUNAINEN

HELSINGIN KAUPPAKORKEAKOULU
Organisaatiot ja johtaminen, pro gradu-tutkielma
Kaisa Uronen

TIIVISTELMÄ

27.11.1997

# OHJELMISTOPROSESSIN KEHITTÄMINEN: TAPAUSTUTKIMUS OHJELMISTON KYPSYYSTASOMALLIN (CMM) KÄYTTÄMISESTÄ


**Tavoitteet**

Tämän tutkimuksen päätavoitteet olivat muodostaa geneerinen viitekehys Capability Maturity Modelin (CMM) hyödyllisyyden arviointiin ohjelmistoprosessin kehityksessä, soveltaa tätä viitekehystä CMM:n hyödyllisyyden arvioimiseen kohdeyrityksessä sekä muodostaa toimintasuunnitelma kohdeyritykselle CMM:n edelleen hyödyntämiseksi yrityksen ohjelmistoprosessin kehityksessä.

**Teoreettinen tarkastelu**

Geneerinen viitekehys muodostettiin perustuen olemassaolevaan ohjelmistoprosessien kehitystä koskevaan kirjallisuuteen. Viitekehys perustuu Plan-Do-Study-Act-malliin. CMM sovitettiin tähän malliin, jotta voidaan kuvata ja arvioida sen pääasialliset vaikutusalueet prosessikehityksessä.

**Tapaustutkimus**

Tapaustutkimus perustui haastatteluihin, työryhmiin ja prosessiarviointeihin kohdeyrityksessä. Näistä lähteistä koottua informaatiota käytettiin yhdessä geneerisen viitekehyksen kanssa kohdeyrityksen tilanteen arvioimisessa ja CMM:n hyödyllisyyden arvioimisessa.

**Pääasialliset tulokset**

Analyysin perusteella vaikuttaa siltä, että CMM soveltuu hyvin toiminnankehityksen kohteiden identifioimiseen sekä toiminnankehitysprojektien tuloksellisuuden mittaamiseen. Sen sijaan malli soveltuu huonommin yksittäisten toiminnankehitysprojektien suunnitteluun. Näin ollen kohdeyritys saattaa tarvita CMM:n lisäksi muitakin malleja toiminnankehityksen tukena.

**Toimintasuunnitelma**

Näiden tulosten perusteella laadittiin toimintasuunnitelma. Suunnitelma suosittelee CMM-perusteisen toiminnankehityksen jatkamista kuitenkin niin, että mallin käyttötapoja edelleen kehitetään vastaamaan paremmin organisaation tarpeita. Lisäksi suositellaan CMM:ään liittyvän koulutuksen jatkuvaa tarjoamista organisaation jäsenille.



**Avainsanat**

prosessikehitys, prosessijohtaminen, ohjelmistoprosessit, CMM

HELSINKI SCHOOL OF ECONOMICS AND BUSINESS ADMINISTRATION
Organizations and management, Master's thesis
Kaisa Uronen

**ABSTRACT**

27.11.1997

# SOFTWARE PROCESS IMPROVEMENT: A CASE STUDY OF USING THE CAPABILITY MATURITY MODEL (CMM)


**Objectives**

The main objectives of this thesis were to formulate a generic framework for assessing the usability of the Capability Maturity Model (CMM) in software process improvement, to use this framework in assessing the usability of the CMM in the case organization, and to formulate an action plan for the case organization for further application of the CMM in process improvement.

**Theoretical Study**

The generic framework was formulated as a synthesis of the existing software process improvement literature. The framework was built on the Plan-Do-Study-Act-cycle, and the CMM was positioned in this cycle in order to describe the main areas of influence of the CMM in process improvement.

**Case Study**

The case study was based on interviews, work groups and process assessments in the case organization. The information collected from these sources together with the generic framework was used to describe and analyze the situation of the case company and for assessing the usability of the CMM in the organization.

**Main results**

Based on the analysis, it seems that the CMM is well suited for identifying objects for process improvement and for measuring the progress of process improvement projects. The model is less suited for detailed process improvement project planning, and thus, the case organization may find it necessary to use other, complementary models as well.

**Action Plan**

On the basis of these findings, an action plan was formulated. The plan recommends continuing with CMM-based process improvement and yet further tailoring and developing the way the model is used, and providing more CMM-related training.



**Keywords**

process improvement, process management, software process, CMM

# Table of Contents

# List of Figures

# List of Tables

# Abbreviations and Definitions

## ABBREVIATIONS

| | |
|---|---|
| **CMM** | Capability Maturity Model |
| **DoD** | Department of Defense |
| **ISO** | International Standardisation Organisation |
| **KPA** | Key Process Area |
| **NTC** | Nokia Telecommunications |
| **PDSA** | Plan-Do-Study-Act |
| **PQP** | Project Quality Assurance Person |
| **R&D** | Research & Development |
| **SCM** | Software Configuration Management |
| **SEI** | Software Engineering Institute |
| **SOP** | Standard Operating Procedure |
| **SPICE** | Software Process Improvement and Capability dEtermination |
| **SQA** | Software Quality Assurance |
| **SW** | Software |
| **SWP** | Switching Platforms |
| **TRILLIUM** | Telecom Product Development Process Capability |

## DEFINITIONS

| | |
|---|---|
| **Institutionalization** | The building of infrastructure and corporate culture that support methods, practices, and procedures so that they are ongoing way of doing business, even after those who originally defined them are gone |
| **Software organization** | An organization developing software products |

# 1 Introduction

## 1.1 Background

The Capability Maturity Model (CMM) is an American software maturity framework developed by the Software Engineering Institute (SEI) of the Carnegie Mellon University. The development of the model was initiated by the US Department of Defense (DoD) in the 1980's. The DoD needed a method for assessing and evaluating its software subcontractors, and the SEI originally developed the CMM for this purpose. The first version of the model was published in 1991, and the current version 1.1 was published in 1993. Since then, many software development organizations around the world have discovered the model. These organizations use the CMM for two purposes: they want to gain a better understanding of their current processes, and they want to have a framework to guide their process improvement efforts.

The CMM is based on the concept of process maturity. A mature software organization possesses an organization-wide ability for managing software development and maintenance processes. A mature process is communicated both to existing staff and new employees, and work activities are carried out according to the planned process. The process is also fit for use, and continuously improved. The roles and responsibilities in a mature process are clear.

The CMM describes an evolutionary path from ad hoc, chaotic processes, to mature, disciplined software processes. The model consists of five process maturity levels:

1. Initial (ad hoc) level
2. Repeatable level
3. Defined level
4. Managed level
5. Optimizing level.

The maturity levels form a hierarchical structure so that the first level is the lowest level of process maturity and the fifth is the highest. Each maturity level (exception for the first) contains a set of process goals that, when satisfied, stabilize an important component of the software process. In order to be at a certain maturity level, a software development organization must fulfill the requirements set by the process goals of that maturity level as well as the requirements of the previous levels.

When using the CMM, the software organization is first assessed against the CMM requirements. The purpose of the assessment is to determine the level of maturity of the software process in that organization. After the maturity level is determined, the organization can start to use the CMM model as a framework for process improvement: the first things to be improved in the software process are the ones in the next maturity level. For example, if an organization is found to be on maturity level 2 (repeatable), it can direct its process improvement efforts to the process areas that are required on maturity level 3. All organizations have, at all times, several possibilities for process improvement, and it is impossible to implement all of them. In order to alleviate this problem, the CMM model is structured so that the issues on lower maturity levels are prerequisites for the successful implementation of the issues on higher levels. Therefore, the model can be used to determine the priorities between different process improvement issues.

## 1.2 Research Problem

The research problem of the thesis can be presented as follows:

"How to successfully use the CMM model in software process improvement?"

This research problem can be divided into two sub-problems:

- What kinds of experiences are presented in the literature on the use of the CMM?
- How should the CMM be used in the case company?

These two fundamental questions have been used as a basis when defining the objectives of this study.

### 1.3 Objectives of the Study

When this study was initiated, the case organization had already been using the CMM for some time. At that point, the organization wanted to get an objective assessment of the usability of the CMM in the organization, and an action plan for further utilization of the model. With these purposes in mind, the objectives of the thesis have been formulated as follows:

1. Presenting an overview of the present knowledge on the use of the CMM in process improvement
2. Presenting an overview of current product and especially software development process improvement issues faced by improvement programs
3. Formulating a framework for assessing the usability of the CMM in software process improvement
4. Presenting the results and experiences from a CMM-based improvement project in the case organization
5. Analyzing the results and experiences, and assessing the usability of the CMM in software process improvement in the case organization
6. Creating an action plan for the case organization for further application of the CMM in process improvement

### 1.4 Scope of the Research

This thesis concentrates on the application of the CMM model. Thus, while other similar models exist, these are only briefly presented and not discussed in detail. Also, as the CMM was originally developed for software development, this thesis does not discuss other types of product development processes.

The CMM model has two generic uses:

- Software process assessments that an organization uses to determine the state of its current software process, to determine the high-priority software process-related issues facing that organization, and to obtain support for software process improvement

- Software capability evaluations that an organization uses to identify contractors who are qualified to perform the software work, or to monitor the state of the software process used on an existing software effort.

This thesis discusses only the use of the CMM in software process assessments. Thus, software capability evaluations are beyond the scope of this thesis. The motivation, objective, outcome, and ownership of the results of these methods differ significantly, and since at the moment, software capability evaluations are not relevant to the case organization of this thesis, they are not included within the scope.

The focus of the case study of this thesis is on the usability and ways of application of the CMM model. Due to confidentiality reasons, the assessment results of the case organization cannot be discussed in detail.

## 1.5 Research Method and Sources of Information

The study is carried out in four stages. In the first stage, software process improvement literature focusing on the CMM is studied and a theoretical framework for the case study is formulated. The second stage consists of carrying out a CMM-based improvement project in the case organization and collecting information about the case organization. The third stage consists of analyzing the results and experiences based on the framework formulated in the first stage, and assessing the usability of the CMM in the case organization. The final stage consists of synthesizing the knowledge and formulating an action plan for further actions.

The study is carried out in close cooperation and team work with the personnel in the case organization.

## 1.6 The Structure of the Thesis

The first part of the thesis discusses the process improvement field in general, current issues in product and software process improvement, and the CMM model and its use in software process improvement. The discussion consists of a literature study and the formulation of a theoretical framework in Chapter 2.

The second part applies the knowledge from the first part to one unit of a Finnish telecommunications company. This case study part consists of:

- The description of the case organization and its software process (Chapter 3)
- The results and experiences from a CMM-based improvement project (Chapter 4)
- Assessment of the usability of the CMM in the case organization (Chapter 5)
- An action plan based on the CMM for process improvement in the case organization. (Chapter 6)

# 2 PRIOR RESEARCH IN PROCESS IMPROVEMENT AND THE CAPABILITY MATURITY MODEL (CMM)

## 2.1 Process Improvement

The last few decades have witnessed a remarkable change in the way quality is perceived in businesses. In the 1950's, inspecting the product was all you needed to do to succeed in the competition. Now, you have to manage and improve your processes just to survive.

The end of World War II started the mass production era. The need for new products was so huge that consumers were happy to get just about anything – complaining about the quality of the products was a luxury few could afford. However, already in the 1950's, the Japanese started to become interested in the teachings of W. E. Deming, an American statistician to whom the Americans would not listen in the early post-war years. He and other quality gurus spoke of statistical quality control, quality planning and management responsibility. They made the Japanese realize that quality is a comprehensive way to manage the whole company, not just a matter of inspections. During the following decades, this realization brought the Japanese to a leading position in quality issues – and at the same time, quality became recognized as one of the most important factors of success in modern day businesses.[1] [2] [3] [4] [5]

The remarkable advances of the Japanese made the West consider the link between quality and business success. Only in the late 1970's and early 1980's did western companies seriously start to adopt the principles of quality

---

[1] Logothetis, N. 1992. pp. xii, 19, 28, 62

[2] Harrington, H. J. 1991. pp. 1 - 3

[3] Pall, Gabriel A. 1987. p. 2

[4] Pall, Gabriel A. 1987. p. 10

[5] Lillrank, Paul. 1991.

management, and focus on quality of operations as well as product quality. This quality movement brought along a new concept that would make it possible to bring quality into everyday life. This new concept was process management.

Pall[6] defines a process as "a logical organization of people, materials, energy, equipment, and procedures into work activities designed to produce a specified end result (work product)". It may be added that an important feature of a process is that the output of the process is of greater value than the input.[7] According to Pall,[8] process management "emphasizes conformance to customer requirements by means of defect-free output in the most efficient and productive manner and the planning and implementation of changes to the process in a timely and orderly way to meet changing requirements and other needs anticipated in the future". The concept of process has been used in manufacturing since the 19th century,[9] but the novel idea in process management is to perceive all activities – whether related to management, manufacturing or for example services – in a company as processes and to use this perception as a basis of management. We will now take a closer look at some aspects in process improvement: different models of process improvement, process improvement supporters and promoters, and the goals of process improvement.

### 2.1.1 Models of Process Improvement

In the process development literature, process management and improvement are presented as a series of steps that should be carried out consecutively in order for process management to bring about concrete results. Some examples of these "step-by-step" lists are presented in the following. Several other authors have presented their models as well, so these lists are presented here just to give a general idea of process management models.

[6] Pall, Gabriel A. 1987. p. 25

[7] Melan, Eugene H. 1992. p. 15

[8] Pall Gabriel A. 1987. p. 25

[9] Melan, Eugene H. 1992. p. 13

Pall[10] divides process management into three stages:

- Commitment to quality: establishing management commitment to quality improvement and implementing the commitment in terms of management infrastructure and process management measures in the shortest possible time
- Managing Quality: consolidating, developing and operationalizing major quality management components as well as organizational, design and measurement aspects along with supporting activities
- Maturity: quality management and process optimization are fully operational, in balance and tied together by common measurements.

Melan's[11] Process Management Model also divides process management into three phases:

- Initialization: assign process ownership, define boundaries and interfaces
- Definition: define the process, establish control points
- Control: implement measurements, obtain feedback and perform corrective action

The Process Quality Management and Improvement Model by AT&T[12] resembles Melan's model, but it puts more emphasis on the identification of improvement opportunities:

- Ownership: establish process management responsibilities, define process and identify customer requirements
- Assessment: define & establish measures, assess conformance to customer requirements
- Opportunity selection: investigate process to identify improvement opportunities, rank improvement opportunities and set objectives

---

[10] Pall, Gabriel A. 1987. pp 195 - 225

[11] Melan, Eugene H. 1992. p. 78

[12] AT&T Process Quality Management and Improvement Guidelines. 1988. p. 15

- Improvement: improve process quality

Harrington's[13] Business Process Improvement Model has five stages that are further divided to several sub-stages. Here we list only the upper level stages but state that this model is more complete than Melan's or AT&T's as it stresses the importance of initializing process improvement (Phase I) more than the others. Also, the phase of continuous improvement is more thoroughly presented in this model.

- Phase I: organizing for improvement
- Phase II: understanding the process
- Phase III: streamlining
- Phase IV: measurement and control
- Phase V: continuous improvement

Dark's[14] Process Improvement Model has ten stages. More than the other models, it stresses the implementation of changes, as the last five stages are all related to it.

- Establish process improvement team
- Define problem
- Collect data
- Set targets
- Analyze performance gap
- Communicate results
- Propose and select solutions
- Plan for implementation
- Implement and test
- Continue to improve

As can clearly be seen, the different models are, in essence, very similar, as they are all variations on the same theme: Deming's famous continuous improvement

[13] Harrington, H. J. 1991. pp. 21 - 22

[14] Dark, Tom. 1994. p. 15

cycle Plan-Do-Study-Act.

The PDSA-cycle is originally based on the work of Walter A. Shewhart, an American statistician. W. E. Deming presented this cycle to the Japanese in the beginning of the 1950's, and the Japanese subsequently started to call it the Deming cycle, while Deming himself calls it the Shewhart cycle. To add to the confusion, the cycle used to be called the Plan-Do-Check-Act-cycle, but Deming changed "check" to "study" in order to emphasize the meaning of studying and learning from experiences. The PDSA-cycle consists of the following activities:

- Plan: plan the change aimed at improvement
- Do: Execute the change
- Study: Study the results – did it work?
- Act: Institutionalize the change, abandon it or do it again.[15]

Each of the process improvement models presented above contains (in this order) the establishment of process improvement organization; the description, the analysis and the measurement of the process to be improved; and the continuous improvement of the process. Thus, the Deming-cycle can be found behind each of them. This is why the theoretical framework of this thesis presented in Section 2.1.4 will be based on the Deming-cycle.

### 2.1.2 Supporters and Promoters

The PDSA-cycle is a very useful way of modeling process improvement. However, in order for process improvement to really work, it needs enablers that in this thesis will be divided into promoters and supporters. Without these enablers, process improvement becomes just another management fad that dies out quietly.

Supporters are passive enablers that prevent the accomplished improvements from being lost again. Promoters, on the other hand, are active enablers that will

[15] See e.g. Deming, W. E., 1986, or Total Quality, 1990

promote further improvements. A common analogy in process improvement is that process improvement is like walking an escalator in the wrong direction. If you stop walking, you will end up where you started. If you walk at the same pace as the escalator, you just stay in the same place. Only if you walk faster than the escalator, you will be able to move forwards. A supporter is a factor that keeps you where you are, and a promoter will take you forwards.

In practice, ISO 9000 audits are an example of a supporter: the organization is audited to see if it fulfills the ISO 9000 standard, and the audit result is either yes, it does or no, it does not.. The audit provides a warning if there are any nonconformities when compared to the standard. On the other hand, different quality award systems, e.g. Malcolm Baldrige National Quality Award (USA) or the European Quality Award are examples of promoters. These models are based on continuous improvement, so they encourage organizations to continuously enhance their performance. The quality award models are not so much a question of yes or no, but a quaetion of how much. In the ISO 9000, an organization either fulfills the requirements or it does not. In the quality award models, an organization aims at improving its rating, but it can never reach maximum points – reaching maximum points would mean that there is nothing left to improve, and this is, naturally, not possible in a real-life organization.

### 2.1.3 Setting Process Improvement Goals

Continuous improvement is a beautifully logical paradigm. However, the problem is that you can never be satisfied. No matter how well improvements are accomplished, management will always demand more. In the long run, this will be very demotivating for the personnel. There is, however, a way to avoid this problem: by setting intermediate goals along the way. Harrington[16] presents process qualification as a method of setting these milestone objectives.

Harrington's process qualification model includes five stages against which the business processes of the company will be evaluated. These stages are presented

[16] Harrington, H. J. 1991 pp. 202 - 216

in Table 1.

*Table 1: Process qualification levels according to Harrington*[17]

| Level | Status | Description |
|---|---|---|
| 6 | Unknown | Process status has not been determined |
| 5 | Understood | Process design is understood and operates according to prescribed documentation |
| 4 | Effective | Process is systematically measured, streamlining has started, and end-customer expectations are met |
| 3 | Efficient | Process is streamlined and more efficient |
| 2 | Error-free | Process is highly effective (error-free) and efficient |
| 1 | World-class | Process is world-class and continues to improve |

By using this kind of process qualification, it is possible to establish improvement objectives according to the qualification levels.

The process management literature contains two slightly different approaches to setting the improvement objectives. For example, from the models presented in the beginning of this section, the AT&T process management model, as well as Melan's process management model, contain the underlying assumption that you first measure and investigate your process to find areas that are not working well. Then you set your objectives based on the results of your investigation. This might be described as a bottom-up approach. Harrington, on the other hand, approaches the problem top-down: he recommends taking a generic process qualification model as the basis, and then setting the objectives based on the model. These approaches put process measurement in a somewhat different position. If a rather strong generalization is allowed, we could say that the top-down approach uses measurement mainly as a provider of feedback, whereas the bottom-up approach sees measurement as a method of setting targets.

When comparing these different methods of target setting, it seems that using

[17] Harrington, H. J. 1991. p.206

some "outside" model in setting targets is recommendable. The bottom-up-method may lead to a correction spiral, where the only improvement ideas come from within the process. Using a qualification model helps in putting things into perspective and seeing larger opportunities.[18]

### 2.1.4 Starting Point

We have now drawn a basic map of the environment we are in. The concept of process has been defined and some process improvement models have been presented. Also, we have discussed the necessity of having process improvement supporters and promoters as well as a process assessment model that can be used in setting targets for the process improvement work and then assessing the realization of these targets. As a summary to this section, Figure 1 presents the starting point of the theoretical framework of this thesis: the PDSA-improvement cycle "rolls" up the process improvement path to reach some preset goals, and in order to maintain the movement, it needs promoting and supporting factors. In the rest of this chapter, our task is to position the CMM model in this framework: what are the main problems to be solved in software process improvement; is the CMM promoting or supporting the improvement; and which parts of the PDSA-cycle does it influence the most and how?

[18] Uronen, Kaisa. 1996. p. 32

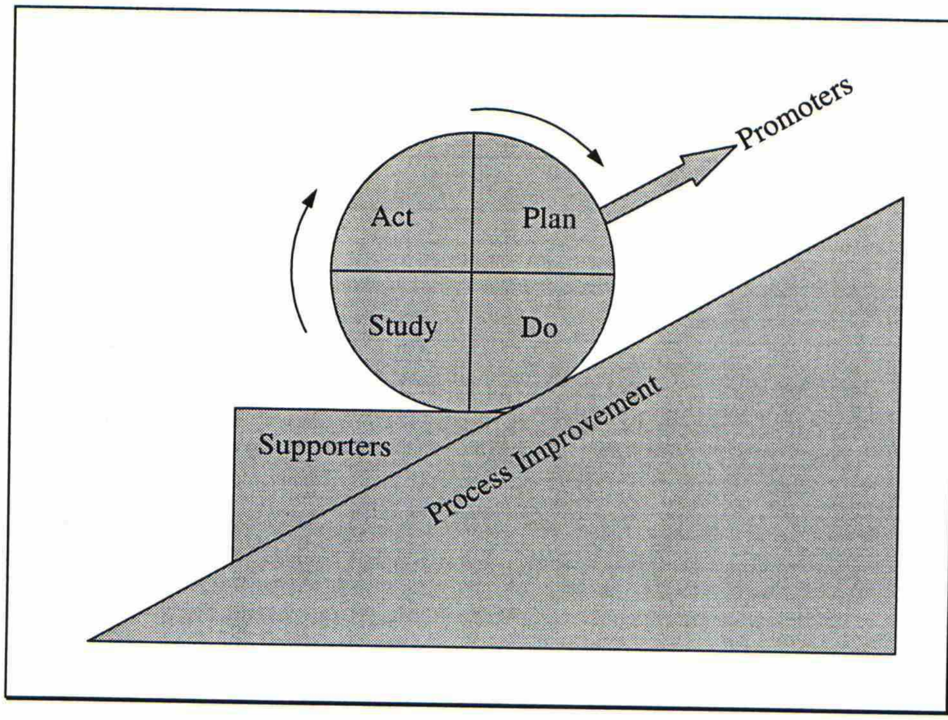


*Figure 1: The Continuous Improvement Framework*

So far, the nature of the discussion has been rather generic: the same principles apply to any kinds of processes. In the next section, we will turn our attention to product development processes and discuss a few issues that currently dominate the discussion in product development process improvement. In Section 2.3, we further focus our attention to a subset of product development processes, namely software development processes. Then, we turn our attention to the CMM. In Section 2.4, the CMM model is described. In Section 2.5, we discuss the use of the CMM in software process improvement, based on the discussion in sections 2.2 and 2.3, and position the model in the environment we have outlined. Section 2.6 discusses the benefits of using the CMM, and Section 2.7 summarizes the literature study.

In Chapters 4 and 5, we then use the knowledge from this chapter to analyze the experiences of the case organization in order to be able to present an action plan for the future.

## 2.2 Product Development Process Improvement

Product development is often considered to be very difficult to manage systematically. There are many reasons for this. Firstly, product development is very information intensive, as both its input and output consist of information. Secondly, product development always contains an element of uncertainty: you can never be certain of the end result in product development, and the schedules and budgets of product development projects can never be totally certain. Thirdly, product development is often very complicated as developing a product means that the processes related to that product need to be developed, as well. It is impossible to develop successful products without considering the processes needed to e.g. produce the product. Wheelwright[19] presents a framework for finding a proper mix of the two. This framework can be seen in Figure 2.

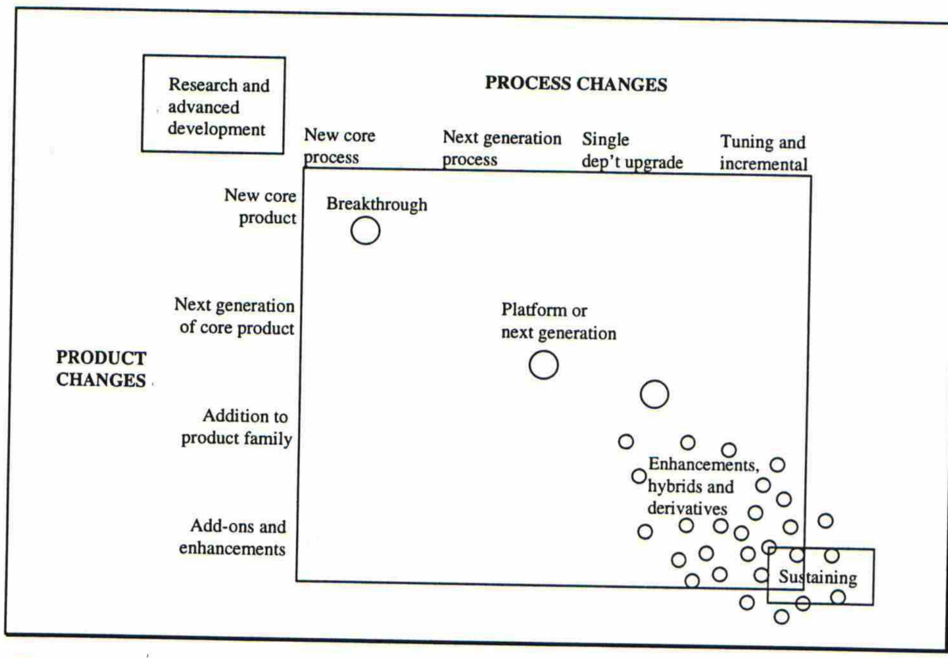


*Figure 2: Product and process development project mix by Wheelwright*

In his model, Wheelwright links product and process changes together and classifies the resulting product-process mixes into five categories according to increasing complexity in both product and process dimensions. Advanced development is the most complex, as it requires extensive innovations in both products and processes. At the other end of the spectrum, sustaining development and small product enhancements contain only minor changes to products and processes.

It is important that a company has product development activities in all

[19] Wheelwright. 1996 (1)

"categories" – e.g. advanced development creates the basis for success in the long term while derivative products create cash flow in the short to medium term.

Product development also poses some special problems in the basics of process management; that is, describing product development processes as well as measuring them has proved to be a difficult task. We will now take a closer look at the description and the measurement of product development processes in order to deepen our understanding of the reasons for the problems in managing product development processes. Then, we will discuss some common problems in product development process improvement, namely product development acceleration and product development teams.

### 2.2.1 Product Development Process Description

Product development has been an object of a lot of research, but there is still a lack of comprehensive models of product development processes. This is mainly due to the iterative nature and inherent complexity of product development.

Wheelwright and Clark have presented a product development funnel model that aims at presenting product development processes at a generic level. The process starts with several new product ideas that are initially developed to see if they could be developed into products. In order to choose among these ideas, the process contains two screenings in which the ideas are compared and assessed to see which of them will be further developed. Finally, the output of the process consists of a few new products that have been refined in the funnel. This model is presented in Figure 3.

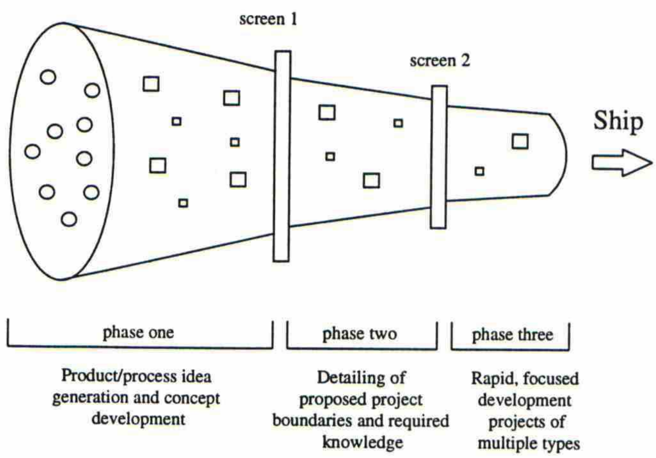


*Figure 3: Product development funnel according to Wheelwright and Clark*[20]

Barclay et al.[21] discuss several attempts at modeling the process, but conclude that most of these models are based on a traditional assumption that product development is a series of sequential stages. The recognition that instead it should be a parallel, iterative series of events has only recently been documented in the literature. Barclay et al. also present a new model – called a sphenomorphic management model because of its wedge-like shape – which aims at taking the nature of product development into account. The model starts with a multitude of product ideas, then continues with the speedy evaluation of these ideas, and, finally, reduces the options until a single, dedicated product is produced. The shape of the model tells about the transition from several ideas to a single realized product. The model is presented in Figure 4.

[20] Wheelwright and Clark. 1992. p. 124

[21] Barclay et al. 1995. p. 356 - 372

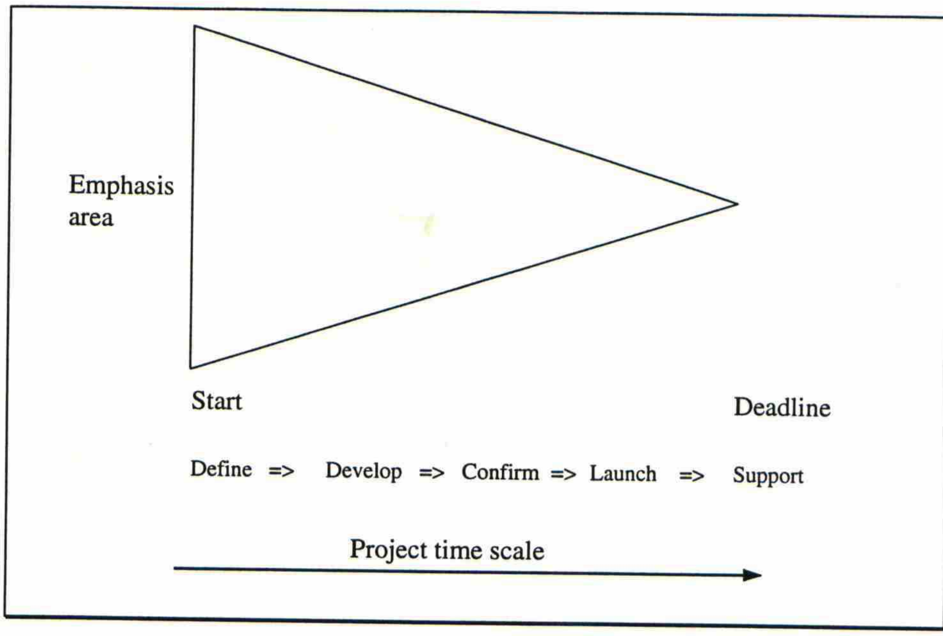


*Figure 4: Sphenomorphic management model by Barclay et al.*[22]

In reality, product development programs consist of many "mini-projects"; false starts, parallel studies, etc. Figure 5 presents a product program that includes several projects, each forming a wedge.

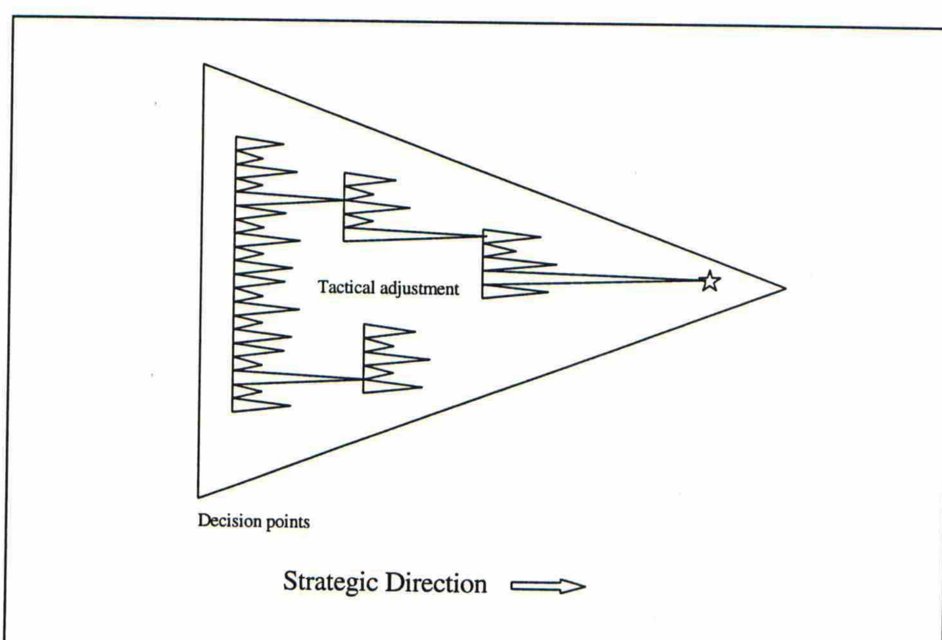


*Figure 5: Project and program sphenomorphs by Barclay et al.*[23]

These generic models presented above do not tell anything about the details of

[22] Barclay et al. 1995. p. 362

[23] Barclay et al. 1995. p.363

specific product development processes. They are, however, useful as a basis for creating the actual description of the development process in a company, as they help in seeing what is relevant and important. Also, they give a generic idea of the things that are to be taken into account in the more detailed flowchart or other descriptions.

### 2.2.2 Product Development Process Measurement

Measuring the overall success of product development efforts has been frustrating because there is no generally recognized metric to measure product development effectiveness. Managers may measure specific projects, but the overall effectiveness of the development process remains unclear. Several attempts have been made to provide such overall measures. For example, McGrath and Romeri[24] attempt to solve this problem by presenting an R&D effectiveness index, which compares the profit from new products to the investment in new product development. The problem, however, is that the results from R&D work may come even many years after the product development efforts have been taken, so the timely measurement of product development process effectiveness remains a problem. Andersin et al.[25] address the issue by presenting measures to be used during a product development project. These include a dynamic cycle time measure and a measure for assessing interaction between different functions. These measures are not completely without problems either. For example, the measure for interaction is based on subjective assessments, which makes it difficult to get reliable information.

We have now covered product development process description and measurement. Next, we focus on two issues that dominate the discussion in product development process improvement; product development acceleration and the organization of product development teams. These are by no means the only problems faced by product development organizations, but they seem to be the most important and common ones at the moment.

---

[24] McGrath et al. 1994; 11. pp. 213 - 220

[25] Andersin, Hans et al. 1994. pp. 15 - 17

### 2.2.3 Product Development Acceleration

In today's turbulent and shifting environment in which companies have to live, time has become a crucial source of competitive advantage. The ability to bring new products to the market faster than competitors brings many benefits:[26]

- Product sales life is extended;
- Market share may be increased;
- Profit margins may become higher;
- Speed creates a perception of excellence among customers.

How can product development then be accelerated? Towner[27] presents four strategies:

- Streamline each stage of development;
- Undertake development activities in parallel;
- Launch the products simultaneously in world markets;
- Release upgrades in product design, support services and business processes after launch.

Reinertsen & Associates[28] list twelve techniques to shorten the cycle time:

- Pre-development activities: economic analysis, managing predevelopment and incremental innovation;
- Development activities: product specifications, product architecture, team selection, team structure, activity overlap, control systems, capacity management, risk management and concurrent engineering.

Further, according to Krubasik and Stein,[29] product development times can be shortened by

- Making the value of time a decision factor;

---

[26] Smith and Reinertsen. 1991. pp. 3 - 6

[27] Towner, Simon J. 1994. pp. 60 - 61

[28] Reinertsen & Associates. 1993.

[29] Krubasik and Stein. 1989. p. 36

- Focusing on eliminating obstacles;
- Instilling a zero-defect mentality;
- Giving the project manager full control;
- In some cases, throwing money at a project;
- Taking the entire product introduction process into account.

As can be seen, several authors have presented their lists of ways of shortening development times. However, despite the numerous formulas and lists offered, there is not much scientific evidence of their effectiveness – most lists are based on speculation or opinion.[30] The results of an extensive study of 103 new product projects by Cooper and Kleinschmidt[31] give some scientific background to the determinants of the development cycle time. The top three time savers in order of importance were found to be:

1. Project organization: use of a cross-functional team which is accountable for the whole project, is dedicated to the project, is led by a strong leader, and has top management support;
2. Up-front homework: effort spent in pre-development tasks (such as initial screening, preliminary technical and market assessments) saved time later on;
3. A strong market orientation: when marketing tasks are carried out proficiently, projects tend to be more time efficient.

The writers also stress the importance of product definition – it should be done just before the "go to development" decision point.

As can be seen from the discussion above, one factor seems to keep coming up when discussing product development acceleration. Whether based on scientific data or experience and opinions, using strong, committed, cross-functional teams is clearly a key to faster product development. This issue is discussed more closely later.

[30] Cooper and Kleinschmidt. 1994. p. 382

[31] Cooper and Kleinschmidt. 1994. pp. 387 - 389, 395

It is necessary to bring a word of warning as to the importance of development acceleration versus other product development targets. Accelerating product development must not be the only focus of management's interest. Smith and Reinertsen[32] suggest that in product development, there are four key objectives that are impossible to be pursued simultaneously; development speed, product cost, development program expense and product performance. Instead, we always face some trade-offs between these objectives. As important as product development acceleration is, management must always find a good compromise among these four conflicting objectives. Actually, Nijssen et al.[33] found in their study that haphazardly implementing numerous methods for accelerating new product development may jeopardize the success of the new product and the company. The product development objectives and the trade-offs[34] are presented in Figure 6.

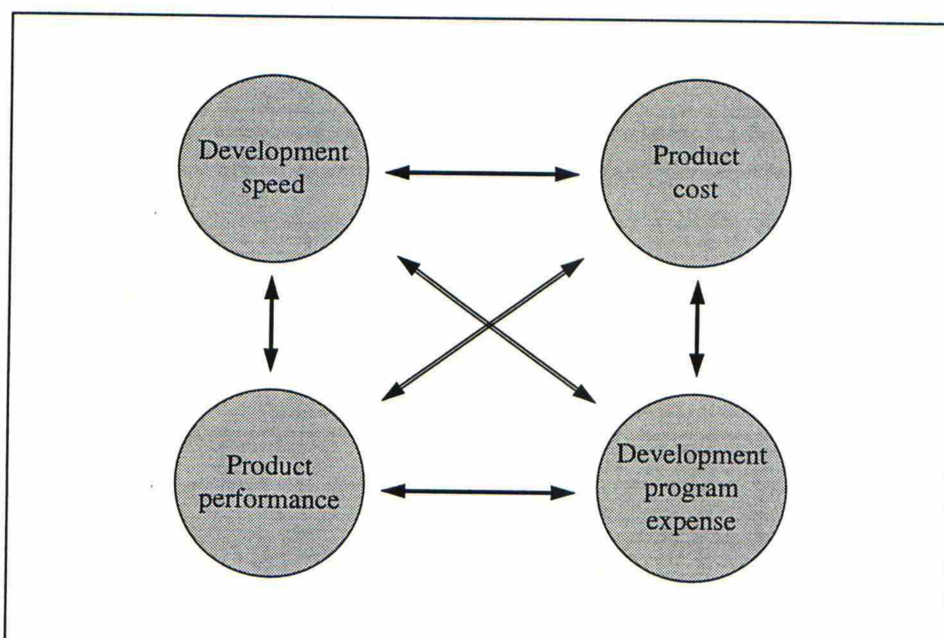


*Figure 6: Four product development objectives and the trade-offs between them according to Smith & Reinertsen*

Now we shall turn to the second burning issue in product development process

[32] Smith and Reinertsen. 1991. pp. 18 - 27

[33] Nijssen et al. 1995. p. 99

[34] Smith and Reinertsen. 1991.

improvement, namely product development teams.

### 2.2.4 Product Development Teams

Wheelwright[35] proposes four different ways to organize product development teams: in functional groups, "lightweight projects", "heavyweight projects" and autonomous project teams. In a functional group, the development is carried out in the functional organization. In a lightweight project, the functional organization controls the resources and decides on the project characteristics, but the project manager coordinates the development. In a heavyweight project, the project manager controls the resources and makes decisions concerning the project, while the functional organization acts mainly as a resource pool. In an autonomous project team, the project team is separated from the functional organization and the project manager has full control. These different types of development teams are presented in Figure 7.

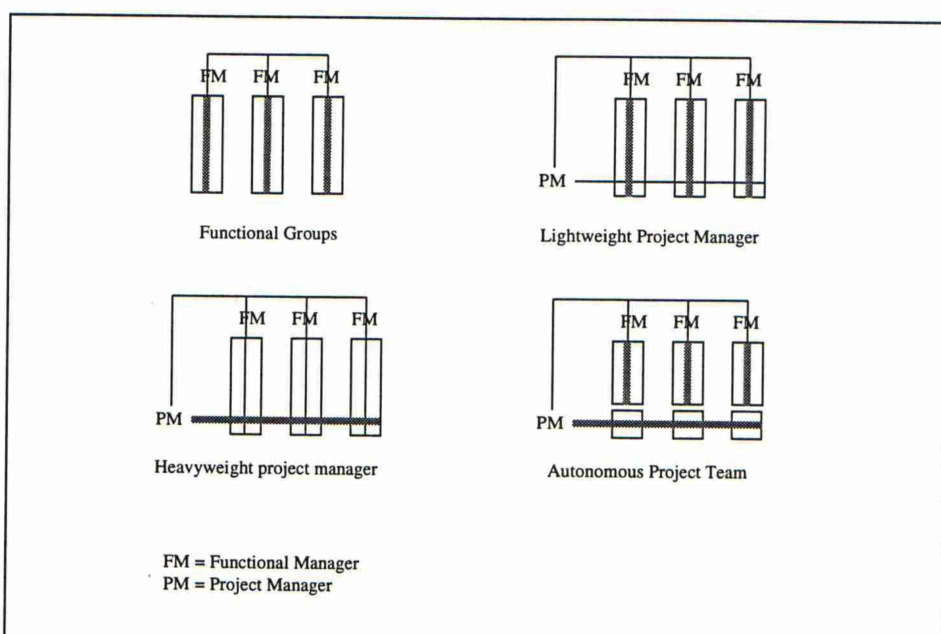


*Figure 7: Four types of product development teams according to Wheelwright*

The appropriate areas of application, strengths and weaknesses of the different team types are presented in Table 2.

[35] Wheelwright. 1996 (2)

*Table 2: Development Teams according to Wheelwright*

| Team type | Application area | Strengths | Weaknesses |
|---|---|---|---|
| Functional | Advanced development and technology infusion | Optimize functional resource utilization, task expertise, depth, control/functional accountability, career path | Do not own business results, lack of breadth, discipline/ bureaucratic, task oriented, slow/codification, disjointed/ mismatches, turf/power wins (technical expertise) |
| Lightweight project team | Derivative products from a generic platform | Improves over functional; adds communication links | Person - lightweight, project focus is weak; gets blame |
| Heavyweight project team | Platforms | Gives project focus, provides commitment/ accountability (core team + leader) | Breadth required, hard to staff, number of people required |
| Autonomous team | Breakthroughs | Focus on results, own business objectives | Independent/ not integrated with others, autonomy is core value |

According to Wheelwright, for most organizations, development efforts are either functionally based or autonomous team based; effective, integrated teams within a functional organization are rare. On the other hand, the literature on product development improvement largely agrees that strong, committed, cross-functional teams are a key to successful product development. But what are these integrated teams like then? The following is a brief summary of what the product development literature says about the characteristics of successful product development teams.[36]

First of all, a product development team must be strong. The team needs a strong leader, it must have top management support, it must have common objectives known to all members and it must have the ability to pursue those targets. The role of the team leader is crucial: (s)he must have enough power and authority to truly lead the group – for a development team leader, visionary and leadership

[36] See e.g. Smith and Reinertsen 1991, Wheelwright and Clark 1992

skills are even more important than technical and management skills.

The team must also be committed to the project at hand. This usually means that the team members must participate in the team full-time, or that they have at most one other project going on. This also means that colocation of the team must be seriously considered. As a last point, commitment implies that the team must be responsible for the product concept and its commercial delivery as a whole.

The third characteristic of a development team is cross-functionality. Outstanding product development requires effective action from all major functions in the company; engineering, marketing, manufacturing and so on.[37] Cross-functional teams are seen as a way to bring separated efforts of the different functions together so that everyone in the company is fighting the same battle. Cross-functional teams also promote concurrent engineering, as a team makes it easier to do things simultaneously instead of sequentially. Thus, cross-functionality brings two kinds of advantages; time is saved as activities happen concurrently, and conflicts can be avoided when different functions know what the others are doing. Even though the interfaces between product development and other functions of a company fall outside the scope of this thesis, it is always important to keep cross-functionality in mind when discussing product development teams.

Building these kinds of strong, committed, cross-functional teams is however not totally without problems. Pelled and Adler[38] present their findings on conflicts resulting from multifunctionality of development teams. They conclude that functional diversity almost always results in task and emotional intergroup conflict. Depending on how the situation is managed, this conflict may be either constructive or destructive. Therefore, it is important that managers learn to analyze the conflict situation and to act accordingly.

Besides conflict within the development group, conflict may arise in the

[37] Wheelwright and Clark. 1992. p. 165

[38] Pelled and Adler. 1994. p. 21 - 27

organization as a whole, as creating development teams leads more or less to a matrix organization. De Laat[39] discusses the conflicts caused by matrix organizations, and concludes that some conflict is bound to arise between project management and line organization. He lists several issues that influence the potential for conflict. Some of his findings are as follows:

- An integrative management style by the project management dampens the tendency for conflict;
- If the general management balances the interests of both arms of the matrix, potential for conflict is minimized;
- If functional management is excluded from strategic decision-making, a source of conflict is introduced;
- Handling of support tasks may cause friction, since a combination of project and support tasks in the departments puts the functional managers in an awkward position, especially when they need to balance off between project needs and external requests for support;
- Finally, the study found a striking dynamic inherent in the matrix management that may contribute to dampening conflicts.

As a summary, cross-functional, strong development teams bring many advantages with them but – especially with careless management – may lead to serious conflict. Therefore, when building development teams, it is important to pay special attention to their management. Otherwise, the advantages of teams may be buried under power struggles and personal disputes.

In this section, we have discussed the characteristics of product development processes and the difficulties in improving them. We will now look at a subset of product development processes that poses very special problems for process improvement.

[39] de Laat. 1994. p. 1089 - 1119

## 2.3 Software Process Improvement

We have seen that product development processes are difficult to manage. But within product development, software processes pose even bigger problems for effective management and improvement. Basically, the same things are important in software development as in any other product development, i.e. shorter cycle times, more accurate measures, concurrent engineering, and so on. However, software is a relatively new and immature industry and in many management issues, it still trails far behind hardware. We will now look at the special characteristics of software process improvement from the point of view of software process measurement, concurrent software engineering and the speed of software development. Software process assessment models will also be discussed.

### 2.3.1 Software Process Measurement

Software development presents its own specific problems in process measurement issues (see also 2.2.2). First of all, the estimation of effort and cost for a software project is still largely a mystery, which makes it difficult to establish the baseline for measures of success. Methods based on lines-of-code, such as COCOMO,[40] have been used with varying success. However, new methods, such as the Function Point Analysis, are developed constantly, and may bring help to effort estimation problems. Secondly, the nature of software makes measurement difficult, since, according to Basili,[41] software data is highly error-prone and generally requires special validation provisions. Thirdly, there is still a large group of software professionals, who think that "software is different";[42] that software is some kind of art form that cannot and should not be measured. These attitudes make it difficult to bring systematical measurement methods to software organizations. And finally, software is a relatively new industry with a tremendous growth rate, so despite huge research efforts in the area, not many

[40] Boehm, Barry. 1981. p. 115

[41] Humphrey, Watts S. 1989. p. 307

[42] e.g. Blackburn, J. D. et al. 1994. p. 5

proven methods exist for software measurement.

### 2.3.2 Concurrent Software Engineering

Concurrent engineering has been a popular idea in hardware development for quite a while now, but in software it is much less used – software processes are still largely sequential in nature. This is largely due to the fact that many of the software process models developed to help in software management are sequential – a good example would be the Waterfall Model, which is the most widely used of the models. However, concurrency is needed in software development for many reasons. For example, sequentiality often leads to efforts spent mostly in the latter stages of development, which, in turn, causes a lot of testing and rework.

Blackburn et al.[43] suggest a methodology for bringing concurrent engineering into software. They approach the issue in growing order of complexity:

- Concurrency within one stage in software development;
- Concurrency between different stages;
- Hardware/software overlap;
- Multi-project or multi-platform overlap.

The application of concurrency differs somewhat between these different levels, and there are limits to the applicability of concurrency. Also, project management becomes much more complicated with concurrent engineering. However, despite all the problems, software has a lot to gain from concurrency in the form of time, effort and money saved, which brings us to the issue of software development speed.

### 2.3.3 Acceleration of Software Development

Blackburn et al.[44] present results of a research that aimed at distinguishing managerial practices that result in faster development speed. According to them,

[43] Blackburn, J. D. et al. 1994.

[44] Blackburn, J. D. et al. 1995.

the factors that contribute the most to development speed are as follows:

- Time spent in the early stages of software development (learning what the customer wants, prototyping): time spent here shortens the cycle time later on.
- Team size: smaller teams are more time-efficient. The exception here are the early development stages; fast developers have proportionally more people working in the early stages.

The researchers conclude that the techniques used by the faster software developers closely resemble the prescriptions proposed for speeding up other forms of product development, and that the similarities between software and hardware development may be more important than the differences.[45]

### 2.3.4 Software Process Assessment Models

The software industry has an interesting dilemma to solve: it is an immature industry, but at the same time, it is growing at a huge rate and desperately needs effective management practices. The software process assessment models form the framework for building and assessing these practices. Before concentrating on the most important and widely used of these models, the CMM, we will take a brief look at the different models that presently exist, in order to create an overview of the field of software process improvement models.

In section 2.1, Harrington's generic model for process qualification was presented as an example of process assessment models. For software processes, several similar models exist. The topic of this thesis, the CMM (Capability Maturity Model) by the Software Engineering Institute is just one of these models. The ISO (International Standardization Organization) model SPICE (Software Process Improvement and Capability dEtermination), and the TRILLIUM (Telecom product development process capability) model are other examples of process improvement frameworks for software processes. All of these models share the same fundamental philosophy; they each include a series

[45] Blackburn, J. D. et al. 1995.

of steps or capability levels, starting from an initial, unstructured "chaos" process and ending in a fully managed, optimizing process. The capability of a software process is determined against these predefined capability levels.

The application of the model in a company begins with an assessment of the software process. When the assessment is carried out, the level of the capability of the process can be determined by comparing the results of the assessment to the capability levels described in the model. After the capability level is determined, the company can use the model to decide and prioritize the improvement actions that are needed to raise the process capability to the next level. There are slight differences between the different models – in the definition of the capability levels as well as in the way the process can move from one level to another. In the CMM, all the requirements for the previous level must be met before the process can move to the next level, whereas in SPICE, a process can fulfill parts of the requirements of the next level even if the previous level is not fully mastered.[46] [47] [48] [49]

The fact that there are several different frameworks causes the need for harmonization between these models. The most important harmonization happens between the CMM and the European ISO standards ISO 9000 and SPICE. The CMM version 2 is currently being prepared, and its release time frame will coincide with the ISO 9000 revision and the first official release of SPICE.[50] All these standards have been compared so as to bring them closer to one another, but the harmonization will not yet be complete. Rozman et al.[51] present the results of a study aiming at defining a quality model that would conform to the requirements of both the ISO 9000 and the CMM. The model has the same Key Process Area structure as the CMM (see 2.4.3) but some Process

---

[46] Dorling, Alec. 1996

[47] SPICE process assessment guide. 1993

[48] Paulk et al, CMU/SEI-93-TR-24

[49] Trillium model overview. 1994

[50] ISPI: Introduction to the Capability Maturity Model, 1997

[51] Rozman et al. 1997

Areas have been revised or added in order to fulfill the ISO requirements. They claim that the integrated model proved to be successful, so it seems that in the future, software organization may be provided an integrated model conforming to both ISO 9000 and the CMM.

We have now discussed product development and software processes, some current problems in developing them and presented an overview of different process improvement frameworks that exist for software organizations. Now we turn our attention to the CMM. Section 2.4 describes the CMM model. In section 2.5, we discuss whether the CMM can be used to solve the problems presented in this and the previous section, and, finally, we position the model in the framework which is presented in Figure 1 on page 14.

## 2.4 Presentation of the Capability Maturity Model (CMM)

### 2.4.1 Background

Software has been a rather undisciplined industry. Proper management methods have been largely missing and software development has been seen as a craftwork where the heroic efforts of individual programmers make the difference. In the 1980's, however, the need for a more disciplined approach began to emerge. For example, the United States Department of Defense expressed a need for a method of assessing the capability of its software contractors. The Software Engineering Institute (SEI) of Carnegie Mellon University took on the challenge, and eventually this led to the publishing of the Capability Maturity Model (CMM) version 1.0 in 1991 and a later version 1.1 in 1993.

An old Chinese proverb says, *"If you don't know where you are going, any road will do"*. To this, Watts S. Humphrey, one of the main developers of the CMM, has added, *"If you don't know where you are, a map won't help"*. It seems to be a common trait for software organization that they do not know the present status of their process or how they should change the process. To help in these problems, the CMM aims at providing software organizations with guidance on how to gain control of their processes for developing and maintaining software and how to evolve toward a culture of software engineering and management excellence. It helps in selecting process improvement strategies by determining the current process maturity, and identifying the issues most critical to software quality and process improvement.[52] The premise behind the CMM is that the quality of the software system is highly influenced by the quality of the process used to develop and maintain it. This premise implies focusing on the process as well as the product.[53]

### 2.4.2 Process Maturity and Capability

A *mature* process, as defined by the CMM,[54]

- Is consistent with the way the work actually gets done
- Is defined, documented, and continuously improving
- Is supported visibly by management and others
- Is well controlled – process fidelity is audited and enforced
- Uses product and process measurement constructively
- Uses technology in a disciplined way.

In a mature process, it is easier for people to effectively develop their potential; improvements are more successful and sustained; and the likelihood of successful introduction of appropriate technology, techniques, and tools increases significantly.[55]

A second important concept in the CMM is *process capability*, which should not be confused with process performance. Process capability implies the range of expected results that can be achieved by following the process. Process performance, on the other hand, is a measure of the actual results achieved from

[52] Paulk et al, CMU/SEI-93-TR-24

[53] ISPI: Introduction to the Capability Maturity Model, 1997

[54] ISPI: Introduction to the Capability Maturity Model, 1997

[55] ISPI: Introduction to the Capability Maturity Model, 1997

following the process. There is a positive correlation between capability and performance, but it is not linear and it depends on many factors - in other words, a good process usually produces good results, but not always. Also, a lousy process just may produce good results if, for example, the organization is blessed with a guru who heroically finishes the job. The CMM focuses on process capability, not performance.[56]

### 2.4.3 Structure of the CMM

The CMM model comprises of 5 maturity levels that are depicted in Table 3.

*Table 3: CMM Maturity Levels*[57]

| **Level** | **Description** |
|---|---|
| 1. Initial | The software process is characterized as ad hoc, and occasionally even chaotic. Few processes are defined, and success depends on individual effort. |
| 2. Repeatable | Basic project management processes are established to track cost, schedule, and functionality. The necessary process discipline is in place to repeat earlier successes on projects with similar applications. |
| 3. Defined | The software process for both management and engineering activities is documented, standardized and integrated into a standard software process for the organization. All projects use an approved, tailored version of the organization's standard software process for developing and maintaining software. |
| 4. Managed | Detailed measures of the software process and product quality are collected. Both the software process and products are quantitatively understood and controlled. |
| 5. Optimizing | Continuous process improvement is enabled by quantitative feedback from the process and from piloting innovative ideas and technologies. |

Each of these maturity levels (exception for level 1) is decomposed into so called *Key Process Areas* (KPA). Each Key Process Area aims at achieving 2-4 *goals* that are considered important for enhancing process capability. In order for the

[56] ISPI: Introduction to the Capability Maturity Model, 1997

[57] Paulk et al, CMU/SEI-93-TR-24. pp. 8 - 9

organization to reach a maturity level, it must fulfill all KPA goals of that level, as well as all previous levels. This is necessary as each maturity level builds on the capability of the previous level. In other words, maturity levels cannot be skipped.

Level 2 Key Process Areas are: Requirements Management, Software Project Planning, Software Project Tracking and Oversight, Software Subcontract Management, Software Quality Assurance and Software Configuration Management. The predominant need at level 2 is to establish effective software project management.

The KPAs at level 3 include: Organization Process Focus, Organization Process Definition, Training Program, Integrated Software Management, Software Product Engineering, Intergroup Coordination and Peer Reviews. Level 3 builds the software process management foundation, as at level 3, processes must be defined, documented and understood.

At level 4, the principles of statistical process control are applied to the software process. The KPAs are Quantitative Process Management and Software Quality Management.

Finally, the KPAs at level 5 are Defect Prevention, Technology Change Management and Process Change Management. Level 5 concentrates on identifying and eliminating chronic causes of poor performance.[58]

The maturity levels and the KPAs are presented in Figure 8.

[58] ISPI: Introduction to the Capability Maturity Model, 1997

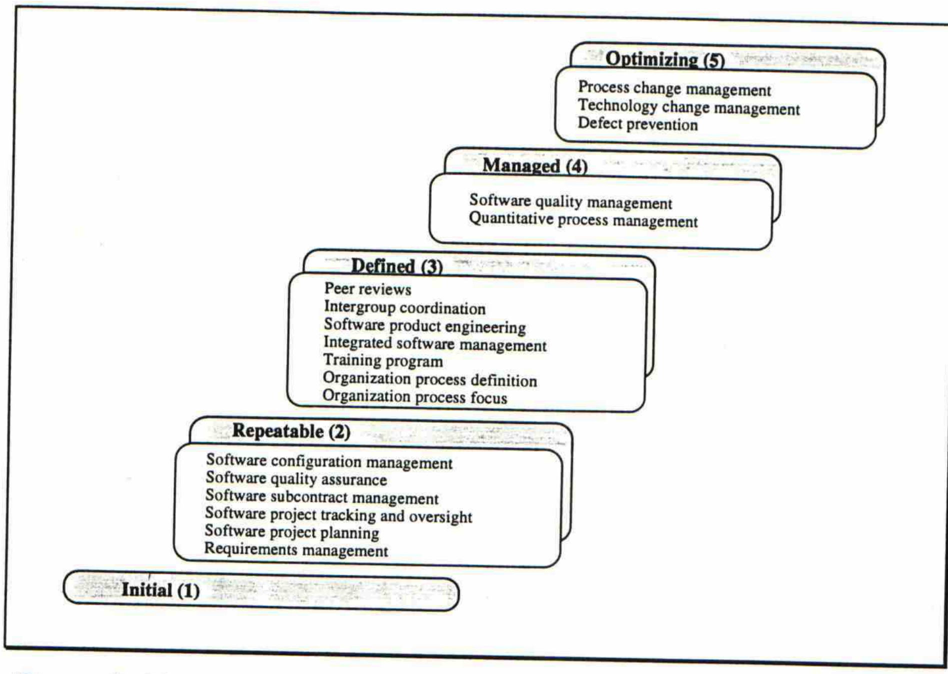


*Figure 8: The Key Process Areas of CMM*

The Key Process Areas are organized by so called *Common Features* which further consist of *key practices*. Figure 9 presents the CMM structure.

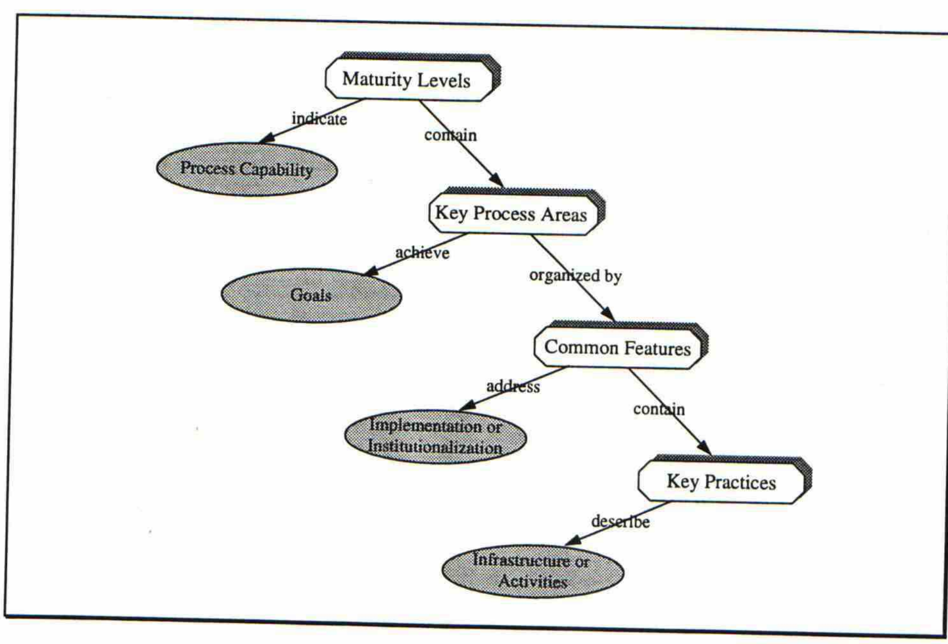


*Figure 9: The CMM Structure*[59]

The Common Features are: Commitment to Perform; Ability to Perform; Activities Performed; Measurement and Analysis; and Verifying Implementation.

[59] Paulk et al, CMU/SEI-93-TR-24. p. 40

- Commitment to Perform describes the actions the organization must take to ensure that the process is established and will endure; Typically, it includes policy and leadership issues
- Ability to Perform describes the preconditions that must exist in the project or organization to implement the software process completely; It typically includes function, resource, delegation, training and orientation issues
- Activities Performed describe the roles and procedures necessary to implement the KPA; Typically, it includes establishing plans and procedures, performing the work, tracking it, and taking corrective actions as necessary
- Measurement and Analysis describes the need to measure the process and analyze the measurements, and it includes examples of possible measurements
- Verifying Implementation describes the steps to ensure that the activities are performed in compliance with the process that has been established; It includes reviews and audits by senior management, project management and software quality assurance personnel.[60]

The practices in the Common Feature Activities Performed describe what must be implemented to establish a process capability. The other practices form the basis by which the organization can institutionalize the practices described in the Activities Performed.[61]

Each Common Feature in each KPA consists of several key practices. These practices describe the fundamental policies, procedures and activities. They focus on what is to be done, but do not mandate how it should be done.[62] This is very important, as the CMM is often falsely interpreted as telling the organization how it should fulfill the goals of each KPA. The CMM only tells what is to be done, not how.[63] When addressed collectively, the key practices of a KPA

[60] ISPI: Introduction to the Capability Maturity Model, 1997

[61] Paulk et al, CMU/SEI-93-TR-24. p. 39

[62] ISPI: Introduction to the Capability Maturity Model, 1997

[63] Wiegers. 1996

accomplish the goals of that KPA.

In summary, the CMM structure is as follows:

- 5 Maturity Levels
    - 2-7 Key Process Areas per maturity level
        - 2-4 Goals per KPA
        - 5 Common Features per KPA
            - 1-15 Key Practices per Common Feature.

### 2.4.4 Using the CMM

The CMM model can be used for two general purposes: assessment of an organization's own software process and capability evaluations of the software processes of the organization's contractors. Here we will concentrate on assessment of the organization's own process.

In general, the assessment proceeds as follows: Firstly, an assessment team is selected and trained. Secondly, representatives from the site to be assessed complete a CMM maturity questionnaire. The maturity questionnaire contains a few questions per each key process area, and it provides a sample of the issues addressed by the CMM. The purpose of the questionnaire is to provide a springboard for the assessors, to give them some preliminary information about the organization as regards the CMM before the actual site visit. The questionnaire is provided by the SEI.

The responses to the questionnaire are analyzed by the assessment team to identify areas where further exploration is warranted. Thirdly, the team visits the site to conduct interviews and to review documentation in order to gain an understanding of the software process followed. At the end of the visit, the team produces a list of findings that identifies the strengths and weaknesses of the organizations software process. This list will later be used as a basis for recommendations for process improvement. Finally, the team prepares a Key Process Area profile that shows the areas where the organization has, and has not, satisfied the goals of the Key Process Areas.[64]

There are a couple of important things to remember when applying the CMM. First, it is intended for a wide range of situations, so the key practices have to be interpreted in light of the organization's business needs, culture, structure and priorities. They must not be obeyed blindly! Second, the CMM is not exhaustive. It contains only key processes and practices, so there are several software management and engineering practices not included in the CMM. Finally, as mentioned earlier, the CMM does not inform on how to aim at fulfilling the goals. It only says what the goals are and what must be done to reach them. For example, the CMM states that a plan must be prepared and maintained for the software project. It does not say what the plan should include, who should make it or how it should be updated.

### 2.4.5 Expected Results and Determinants of Success

We have now described the CMM model and how it should be used. Next, we will look at what kind of returns can be expected when using the model, and what kind of prerequisites are needed for reaching these results.

Some studies exist on the results of CMM-based process improvement. For example, Table 4 presents the results of a study by the Software Engineering Institute that investigated the influence of using the CMM model in software process improvement.

[64] Paulk et al, CMU/SEI-93-TR-24. pp. 44 - 47

*Table 4: Results from using CMM in software process improvement*[65]

| Category | Range | Median |
|---|---|---|
| Total yearly cost of software process improvement (SPI) activities | \$49 000 - \$1 202 000 | \$245 000 |
| Years engaged in SPI | 1 - 9 | 3.5 |
| Cost of SPI per software engineer | \$409 - \$2004 | \$1375 |
| Productivity gain per year | 9 % - 67 % | 35 % |
| Early detection gain per year (defects discovered pre-test) | 6 % - 25 % | 22 % |
| Yearly reduction in time to market | 15 % - 23 % | 19 % |
| Yearly reduction in post-release defect reports | 10 % - 94 % | 39 % |
| Business value of investment in SPI (value returned on each dollar invested) | 4.0 - 8.8 | 5.0 |

Further, McGuire[66] presents the results of an empirical study on an experienced software development team that started a CMM-based process improvement program. Four different surveys were carried out in the team. The first was a survey on the team members' perceptions of the strength of a wide variety of factors related to the existing software development environment. The factors were categorized under eleven classes; team background, teamwork behaviors, team leadership, change management, quality focus, process focus, customer focus, project management methodologies, metrics and tools. Two months later, the team was surveyed on their perceptions concerning the extent at which various change related issues were encountered during the initial stages of process improvement. At the same time, the team members were asked to evaluate the extent to which they observed various change management strategies being employed. The fourth survey was carried out six months later, and the questions were the same as in the first survey. A five-point scale was used on all surveys, where 1=none, 2=weak, 3=average, 4=strong, 5=excellent.

[65] Herbsleb et al. 1994. p.15

[66] McGuire, E. G. 1996

As the first and fourth surveys are the most interesting from the point of view of this thesis, their results are presented in Table 5. It is interesting to see that during the six months between these surveys, all factors had improved and some even considerably.

*Table 5: Ratings of Project and Process Management Issues*

| Category | Before | After | Change |
|---|---|---|---|
| Tools | 2,87 | 3,12 | +0,25 |
| Team Leadership | 2,64 | 3,36 | +0,72 |
| Customer Focus | 2,60 | 3,12 | +0,51 |
| Metrics | 2,60 | 2,73 | +0,13 |
| Team Background | 2,53 | 2,87 | +0,34 |
| Methodologies | 2,31 | 2,75 | +0,44 |
| Project Management | 2,17 | 3,89 | +1,72 |
| Teamwork Behaviors | 1,87 | 2,93 | +1,06 |
| Process Focus | 1,76 | 3,48 | +1,72 |
| Quality Focus | 1,74 | 2,89 | +1,15 |
| Change Management | 1,63 | 3,59 | +1,96 |

Lowe and Cox[67] present the results from a study in the Software Engineering Systems Division of Hewlett-Packard. The Division started applying the CMM in September 1994. According to them, the key results and benefits from CMM-based process improvement were as follows:

- Reduced cycle time (from 18-24 months to 14 months)
- Schedule slip reduced to zero
- Improved execution of projects

[67] Lowe, D. et al. 1996

- Customer orientation
- The ability to respond to changes.

Lowe and Cox conclude that the improvement progress was very successful, and that the CMM provides an excellent framework for defining software engineering process improvement even in a smaller organization.

All these results seem quite impressive, but there are opposing opinions. The CMM model, as well as many other models like the ISO 9000, lacks an explicit connection to the business goals of the organization in question, so it is difficult to predict the exact usefulness of an improvement project based on these models. Therefore, when the organization is considering a CMM-based software process improvement project, many managers are not ready to allocate resources to a project, the usefulness of which they cannot directly see. Reiblein and Symons[68] claim that the main reason for this is that Information Technology managers proposing Software Process Improvement (SPI) projects and the company managers speak a different language – and so fail to find a compromise. In order to link business thinking to SPI, they present a method for the establishment of a process improvement profile which starts with business goals. The method is thus directly connected to the company board's way of thinking, yet it is also compatible with the SPICE-model. The authors state that with minor modifications, the same method is applicable also in CMM-based process improvement.

Another thing to consider is that the rewards of process improvement are not automatic – applying a process improvement framework is not in itself enough. All process improvement requires hard work and dedication in order to succeed. Some factors influencing the success are:[69]

- Senior management support. The importance of this factor cannot be overemphasized. Process improvement brings along so many and so

[68] Reiblein et al. 1997. pp. 89 - 98

[69] Hannus. 1993. pp. 266 - 267

considerable changes that without senior management support, it is simply impossible to introduce them to the organization;

- Clear justification and communication of the changes;
- Management and incentive systems;
- Cross-functional participation;
- Creativity;
- Focused, profound and systematical analysis of the situation;
- Clear responsibilities;
- Fast results. This is also very important. If process improvement does not produce any visible results relatively quickly, it is soon discarded as just another management fad.

This is by no means an exhaustive list of prerequisites of success. However, no list of success factors could be complete – in the end, everything depends on the people who make the difference.

## 2.5 CMM and the Plan-Do-Study-Act -cycle

In the previous section, we described the CMM model, its application and some possible returns from using it. We now return to the beginning of this chapter, where we outlined the basic framework for this thesis: the Plan-Do-Study-Act cycle with promoters and supporters rolling up the hill of continuous improvement. Using the literature we have discussed so far, our aim now is to place the CMM in this framework. In this section, we analyse the role of the CMM in the PDSA-cycle: does it aid planning, doing, studying or acting and how? In Section 2.6, we discuss the CMM's role as a process improvement supporter and promoter, and see how the CMM relates to the common problems in product and software development outlined in Sections 2.2 and 2.3.

The CMM is based on principles of statistical quality control and product quality presented and elaborated by W. Edwards Deming, Walter Shewhart, Joseph

Juran and Philip Crosby, and the PDSA-cycle is one of the basic premises of the model.[70] Let us now take a closer look at the relationship between these two models.

### 2.5.1 PLAN: CMM in Planning Improvements

When planning improvements for an existing process, the CMM provides a lot of help, as it establishes a set of publicly available and largely recognized criteria describing the characteristics of a mature software organization.[71] A software engineering process group equipped with the knowledge of the software process issues and business environment of a particular organization can use the CMM to compare their current practices against the goals of the Key Process Areas of the CMM. The creators of the CMM model remind that the goals should be examined in relation to corporate goals, management priorities, the level of performance of the practice, the value of implementing each practice to the organization, and the ability of the organization to implement a practice in light of its culture.[72]

This comparison will provide the process group with an understanding of what the main strengths and weaknesses of their software process are. This will further ease the selection and prioritization of the improvement initiatives to be implemented.

However, when planning the improvement initiatives in detail, the CMM does not help much. It does say what should be accomplished but it does not say anything about how to achieve it. Therefore, the process group must itself determine which process improvements are needed the most, how to effect this change and how to obtain the necessary buy-in. The CMM does provide a starting point for discussion about process improvement by helping to surface different assumptions about commonly accepted software engineering

---

[70] Paulk et al, CMU/SEI-93-TR-24. pp. 5-6

[71] Paulk et al, CMU/SEI-93-TR-24. pp. 43

[72] Paulk et al, CMU/SEI-93-TR-24. pp. 49

practices.[73]

An important thing to remember is that when starting to use the CMM, it must first be understood in detail. In order to be able interpret the requirements in the context of the organization, the people using the CMM must be thoroughly familiar with the terminology and concepts used in the model.[74] It must be said that the CMM is not always easy to understand since its terminology sounds like legal language. Therefore, some help from consultants may be necessary.

### 2.5.2 DO: CMM in Implementing Improvements

Implementing the chosen improvements is very weakly supported by the CMM. As has been stated, the CMM model intentionally refuses to say anything about how to implement the changes. There are some examples of possible ways of implementation included in the CMM, but these are at a very high level. For example, when the CMM says that the organization must use measurements to determine the status of the planning activities, it does list examples of possible measurements; completion of milestones for the software project planning activities compared to the plan; and work completed, effort and funds expended in the software project planning activities compared to the plan. As can be seen, there is still a lot of flexibility in how the organization decides to define these measurements.

The organizations using the CMM may originally see this refusal to discuss the "how" as a weakness of the model, as this forces the people in the organization to "think themselves". However, this actually is one of the strengths of the model. It is generic enough to be applied in a multitude of situations and over a period of several years or even decades. Also, as the organization is forced to interpret the model, using it will gradually teach the organization to build processes that are suited exactly to its own needs. This is why the CMM is not a "silver bullet": it is not an automatic answer to all the problems the organization has but instead it is

[73] Paulk et al, CMU/SEI-93-TR-24. pp. 49

[74] Lowe et al. 1996

framework within which the organization can address software management and engineering issues.[75]

### 2.5.3 STUDY: CMM in Studying Results

For measuring, studying and analyzing the results of the improvement project, the CMM does provide support and yet the support is at a rather high level. For assessing the success of a single project, the CMM is not suited. However, the model was created to be a basis for process assessments and it contains the five-step maturity framework that in itself helps the organization assess the success of its improvement projects as the organization reaches higher maturity levels. Therefore, the CMM can be used as a high level measurement of the progress within process improvement, but the time frame of the assessment should be several years. For individual improvement projects, the organization must find other measurements.

### 2.5.4 ACT: CMM in Corrective Actions

After the study-stage of the PDSA-cycle, the organization should act based on the information collected about the improvement project. Usually, this means performing some corrective actions in order to correct any deviation between what was planned and what was actually realized. Here again, the support from the CMM is rather weak. The CMM can be used as a reminder of the goal, i.e., what the organization is aiming at. However, it cannot be used to determine how the goal can be reached.

### 2.5.5 Conclusions: CMM and the PDSA

As a conclusion to our discussion of the CMM and the PDSA-cycle, we can say that the CMM is most useful in the PLAN stage. It also provides some assistance in the STUDY stage, but only very little support for the DO and ACT stages.

To present the relationship of the CMM and the PDSA from another point of

[75] Paulk et al, CMU/SEI-93-TR-24. pp. 51

view, we look at a PDSA-based process improvement model called IDEAL that has been launched by the SEI, the developer of the CMM. The IDEAL model consists of five stages that are further divided into substages. The model is presented in Figure 10.

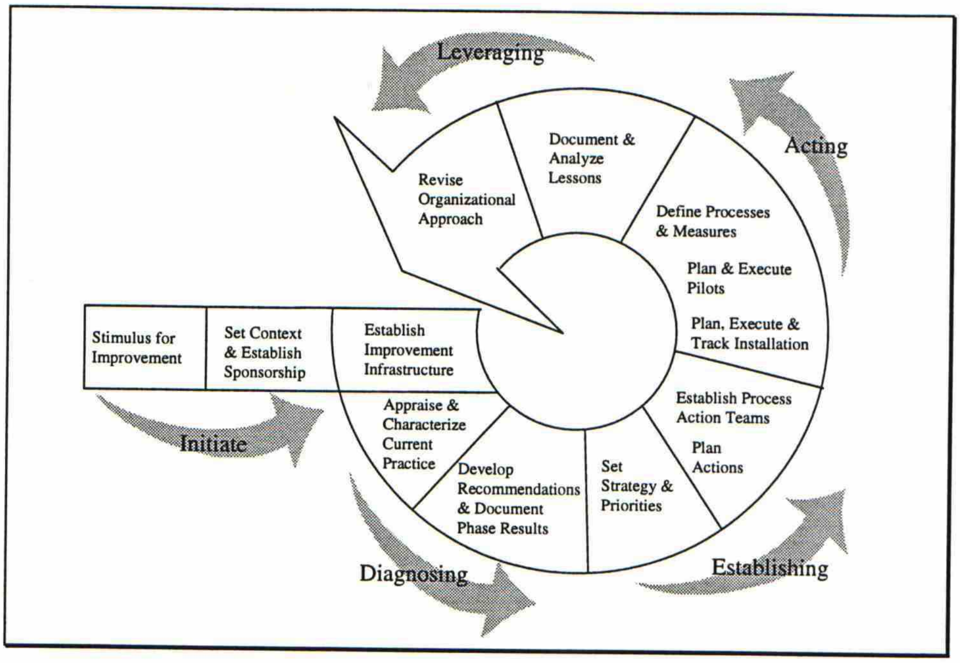


*Figure 10: IDEAL model by the SEI*

Using the terminology of the IDEAL model, we can summarize the relationship between the CMM and continuous process improvement. In the Diagnosing step, the CMM has a crucial role, as it represents a standard against which to determine the weaknesses and strengths of a process. Also, in the Establishing step, the CMM has a central role as contains a set of process goals to be implemented by a program of continuous improvement. In other stages, the CMM is less useful. Therefore, successful process improvement cannot be based on the CMM alone.[76]

## 2.6 CMM's role in Software Process Improvement

Now we have positioned the CMM in the PDSA-cycle. In order to completely position the CMM in the framework of this study, we now discuss two futher

[76] Dymond, K. 1988. pp 1.8 - 1.9

aspects: How the CMM supports or promotes software process improvement, and how it relates to the problems commonly present in software process improvement.

### 2.6.1 CMM: a Supporter or a Promoter?

In the beginning of this chapter, we discussed the concepts of promoters and supporters. The promoters will always promote new process improvements while the supporters will maintain the improvements that have already been achieved. To properly place the CMM in the framework of this thesis, we will now discuss its role as a promoter or a supporter.

The CMM can be seen as both a supporter and a promoter. From the support point of view, the 5-step maturity model supports process improvement since repeated assessments will reveal possible declining in process status. Therefore, using the CMM systematically and over a long period of time will prevent slipping back from achieved improvements. But the maturity model also provides a promoting aspect: a clear, evolutionary path where the next target is never too far away to be reached within an appropriate time frame. The CMM provides convenient milestones so that the organization will get a feeling of achievement when trying to reach a level 5 process. Having too ambitious targets may kill the enthusiasm, but having several consecutive targets will promote the work. A suitable analogy would be building a house. If you start building all parts of the house at once, you will get nothing done, but if you first build the foundation, then the walls and finally the roof, you will be able to finish the job.

### 2.6.2 Solving the Common Problems of Software Processes

As a last step in positioning the CMM in the framework of this study, we now turn our attention to the common software and product development process problems presented in Sections 2.2 and 2.3. These problems were

- Process description
- Process measurement

- Process acceleration
- Organizing product development
- Concurrent software engineering

The CMM is a model that is to be used in process improvement. So how does the model relate to these specific process improvement problems?

When discussing specific process problems, we soon run into the fact that the model is by no means a sufficient tool in itself. It does not provide answers to detailed problems, it only presents a model that, when properly implemented in a real-world organization, should help the organization to solve any problems it may encounter. However, some of the aforementioned problems are better addressed by the CMM model than some others.

Process measurement and process acceleration are the kinds of problems to which the CMM provides some help. Process measurement is especially well presented in the CMM, as the measurement of both the product and the process is an integral part of each KPA. The measurements concerning processes are presented in the Measurement and Analysis Common Feature, while product measurements are listed in Activities (e.g. the requirement for software size measurement in the Software Project Planning KPA is presented in Activity 9). Measurements in the CMM evolve throughout the maturity framework:

- At level 1, measurements are haphazard and its difficult to collect reliable data
- At level 2, projects collect management data about cost, size, effort, schedule, etc.
- At level 3, measurements are consistently defined throughout the organization, and both management and quality data is collected
- At level 4, data analysis is based on the principles of statistical process control, and actual measurements are compared to expected values of mean and variance
- At level 5, continuous improvement is based on business objectives and cost-

benefit analysis.[77]

It is interesting that when implementing measurements "the CMM-way", the original requirements at level 2 are not really too hard to realize in just about any organization. However, when many software organizations consider measurements to be difficult to implement, there is reason to ask if these organizations are trying to build a house starting from the roof?

Process acceleration is not quite as directly addressed in the CMM. However, as the CMM forces an organization to first create solid project management practices and then create equally solid processes that are applied throughout the organization, it is easy to see that process acceleration is then considerably easier. After these project and process practices are implemented and institutionalized, it is just a question of how the organization wants to direct its operations. If process acceleration is a crucial target, good management practices make it easy to realize.

Process description, development team organization and concurrent development are, unfortunately, much less helped by the CMM model. Naturally, when an organization gets its processes under control, all kinds of problems are easier to solve, but the CMM will not provide any further assistance in solving these problems.

We have now positioned the CMM in our theoretical framework, and it is time to summarize this discussion by completing the framework of this study.

## 2.7 Summary of the Theoretical Framework

We have now concluded the discussion about the existing knowledge in the field of software process improvement. In Section 2.1, we outlined the picture by discussing process improvement in general. In Sections 2.2 and 2.3 we focused on product development and further software development processes, and discussed some common problems regarding these processes. Section 2.4

[77] ISPI: Introduction to the Capability Maturity Model, 1997

presented the CMM model, Section 2.5 connected the CMM to the fundamental process improvement model PDSA, and Section 2.6 discussed CMM's role as a process improvement enabler and analyzed the possibilities of using the CMM in solving the problems presented in 2.2 and 2.3.

Now it is time to summarize this discussion. In Figure 11 we have taken the basic framework presented in the beginning of this chapter and added to it the components that were found during this chapter.

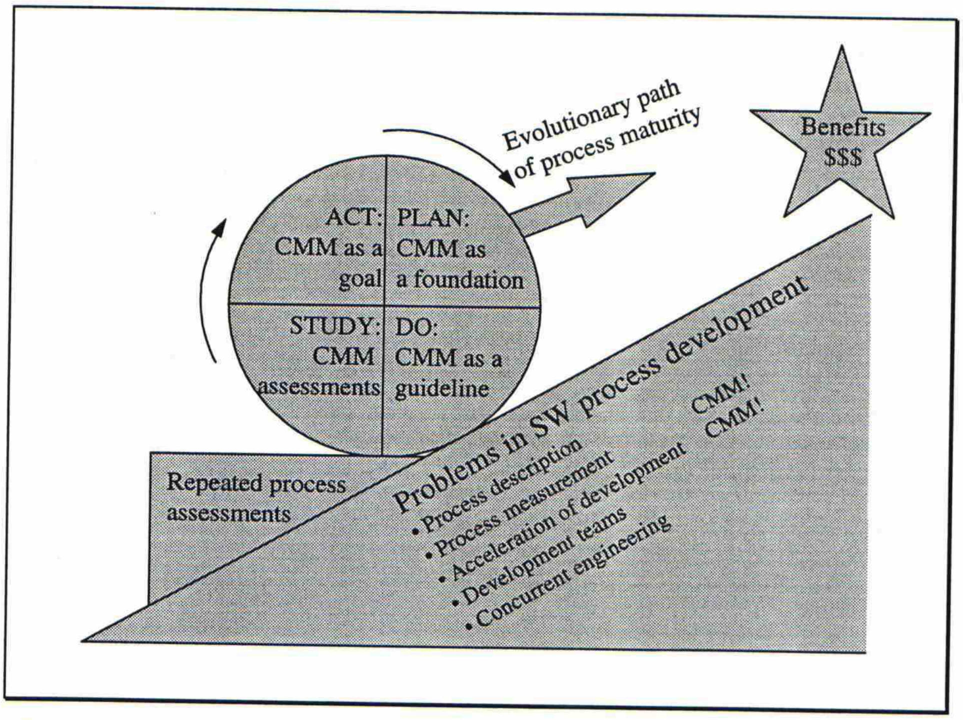


*Figure 11: The Theoretical Framework of This Thesis*

The "process improvement hill" now contains the product development and software process problems we discussed. These are, however just some possible problems; each organization should define itself what its main problems are, and then what questions process improvement tries to answer in that particular organization. The main thing is that the organization has a clear picture of what it tries to achieve with process improvement.

The different roles of the CMM have been added to the PDSA-cycle. In the planning phase, the CMM provides a foundation for the plans, as it describes the goal – a mature process. In the implementation phase, the CMM is somewhat in the background, but it must guide the work so that the different process improvement projects all work towards a common goal. In the studying phase, CMM assessments provide a way of following the development of the process improvement work by measuring the changes in process maturity. In the corrective action phase, the CMM functions again as a reminder of the goal: what it is that should be achieved.

The role of the CMM as a supporter and promoter has also been added to the picture. Repeated process assessments support the work as they will provide an alarm if some achieved improvements are in danger of deterioration. On the other hand, the five-stage evolutionary model promotes process improvement. It stimulates and eases the work by giving intermediary targets instead of just describing the properties of a mature process. The organization is given a target as well as a path to that target, instead of just describing the target.

Finally, the star at the top of the hill is a reminder of what process improvement is all about; getting more value out of the work. Process improvement must give some kind of added value, be it financial, psychological or other. If no value is added, the process improvement work is futile.

We now turn to the case study. In the next chapter, the case organization is presented. In Chapter 4, we present the results and experiences from using the CMM in the case organization, and assess the usability of the CMM model based on the discussion of this chapter.

# 3 R&D in NTC/SWP

## 3.1 Nokia Telecommunications

### 3.1.1 Organization and Products

Nokia Telecommunications (NTC) is a business group of the Nokia Corporation. NTC develops and manufactures telecommunications systems and equipment for both fixed and mobile telephone networks. A fixed network typically consists of trunk and access exchanges, with transmission systems in between the exchanges. A mobile network consists of a mobile switching center, base station controllers, base stations and transmission systems. In NTC, the switching systems of both fixed and mobile networks are based on the DX 200 digital exchange.

The organization of NTC is divided into five divisions:

- Network Systems that is responsible for exchange, intelligent network, network management and private mobile radio businesses as well as defining and delivering NTC-wide systems
- Radio Access Systems that is responsible for base station, base station controller, wireless transmission and fixed wireless businesses
- Fixed Access Systems that is responsible for fixed network and dedicated network businesses
- Information Networking Systems that is responsible for fixed and wireless data and Internet system businesses
- Customer Services that takes care of customer service, e.g. telecommunications network planning, installation and maintenance, as well as customer training and technical support.

The divisions are further divided into Product Lines. The organization chart is presented in Figure 12.

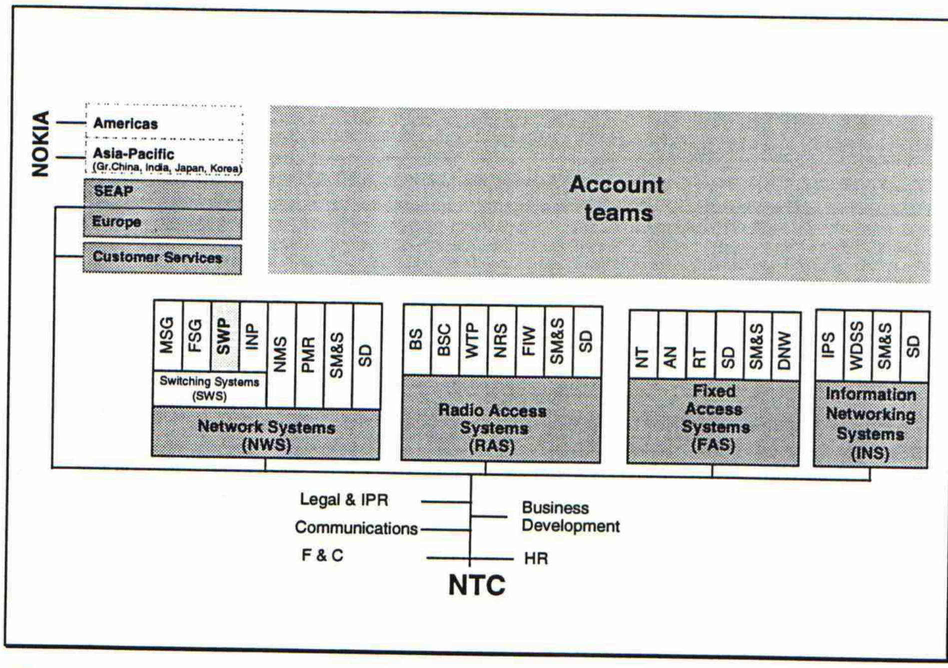


*Figure 12: Nokia Telecommunications Organization*

The object of the case study in this thesis, the Switching Platforms R&D Unit (SWP/R&D), is part of the Network Systems division. SWP/R&D develops a generic platform for the DX 200 exchange that other DX 200-based Product Lines use as a basis for their customer applications. The structure of the DX 200 exchange is presented in Figure 13.

The architecture of the exchange is layered: at "the bottom", there is the generic platform that is divided into Computing Platform and Switching Platform. The Product Lines build their country or customer specific applications "on top" of this platform.

When delivering a new DX 200 exchange or updating existing exchanges, the primary unit to be sold is a feature. A feature consists of software, hardware and documents. Features are not, however, delivered individually, but are combined into packages of features called releases. A release is the primary unit to be delivered.

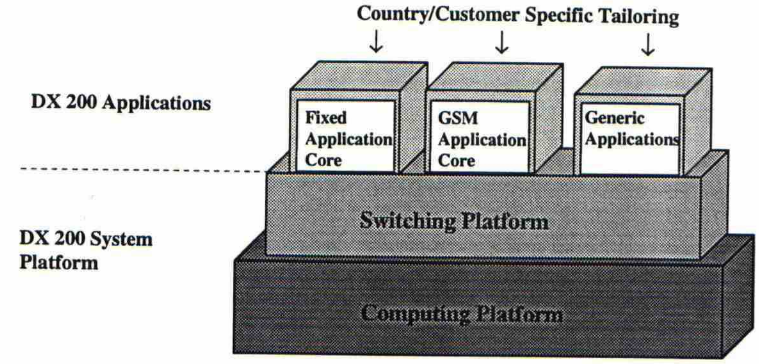


*Figure 13: The structure of the DX 200 exchange*

### 3.1.2 Processes in NTC

The application of process management principles in Nokia started in Nokia Mobile Phones (NMP) in the beginning of the 1990's. It helped NMP obtain some remarkable results, so the other Nokia units started to become interested in process issues as well. After Jorma Ollila, the CEO of Mobile Phones, became the CEO of the whole Nokia Group, process thinking spread even more.

In NTC, process thinking started gradually during the early 1990's. The first big step was the definition of the core processes, the Product Process, the Customer Commitment Process and the Support Processes. These processes are the basis of work in NTC: the Product Process develops telecommunications systems that are then implemented, sold and delivered to the customer by the Customer Commitment Process. The Support Processes do not directly produce added value to the customer but support the core processes. The processes are described in Figure 14.

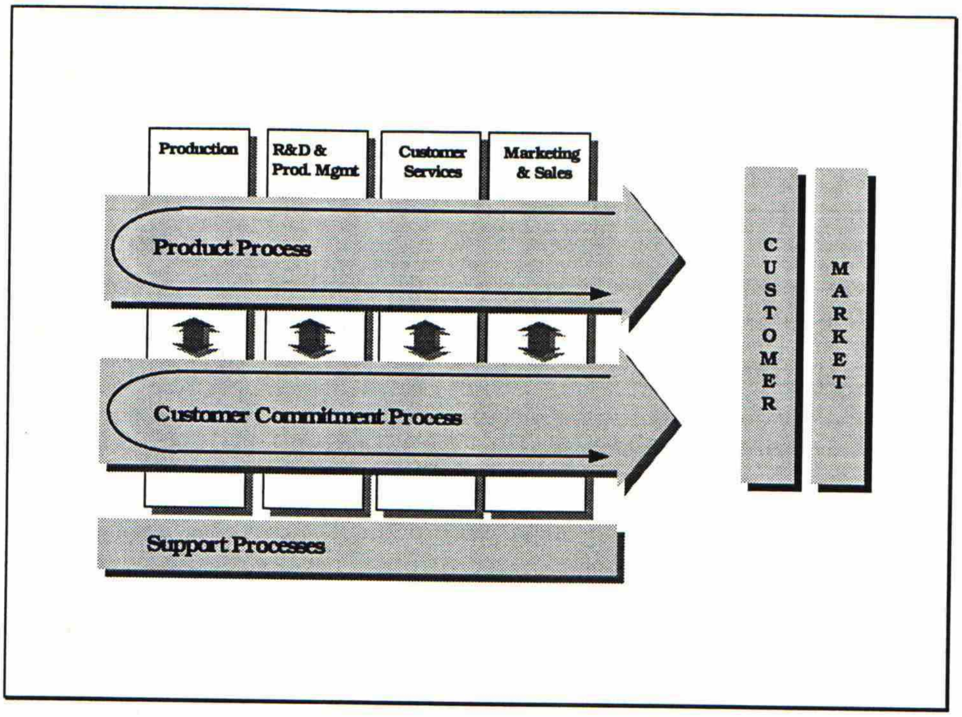


*Figure 14: NTC core business processes*

The Product Process was quickly defined further, and, initially, the goal was to define a common Product Process for the whole NTC. Soon, however, it was noticed that this goal was simply impossible as the Business Units differed too much from one another. The next step was to create a common framework within which the Business Units could then define their own product processes. This goal was reached with a Milestone Management Model that is a standard, phased product development model. The definition of the Customer Commitment Process and the Support Processes has proceeded much slower.

Currently, processes are defined and managed on a Business Unit level, so top management of NTC does not participate in day-to-day process management. The basic building blocks of process management, such as the division of the Product, Customer Commitment and Support Processes, and the Milestone Management Model of the Product Process, were decided at the top management level of NTC. Within the limits of these basic issues, the Business Units are quite independent in their process management activities. Thus in this thesis, senior management refers principally to the top management of SWP and the SWP/R&D, as this is the level of management that mainly influences process management.

## 3.2 R&D in Switching Platforms

### 3.2.1 Line Organization and Products

The R&D Unit in Switching Platforms is organized as a matrix. One dimension of the matrix consists of eight competence area based line organization departments including four DX 200 software development departments and a software department developing tools and methods for the DX 200 departments.

The other dimension consists of five specification and steering departments that control product management, project management, product architecture, long-term research and quality issues throughout the line organization departments. The R&D is mainly situated in the Helsinki area apart from one unit in Äänekoski that is a part of one of the software departments.

On a historical note, the division into four software departments has been done quite recently. The R&D used to have two departments, but each of them was split into two departments in the beginning of June 1997. Thus, during the CMM assessments discussed in Chapter 4, the organization had only two DX 200 software departments.

The R&D unit develops generic platform releases for the DX 200 exchange that integrate new software, hardware and documentation to the previous platform releases. The platform releases serve as a basis for the customer applications developed by other Product Lines of NTC.

The development of a new release in the R&D Unit starts with the collection of customer requirements and the definition of the feature contents of the new release. The product management department is responsible for these activities. After the product definition is completed, the project management department is responsible for coordinating the development projects. The actual project work is performed in the line departments. Software dominates the development, as it accounts for approximately 80 % of all development effort.

### 3.2.2 SWP Product Process

The product development activities are performed in the Product Process. The

SWP Product Process consists of four subprocesses: Product Definition, Product Development, Delivery Capability and Management. This division is illustrated in Figure 15.

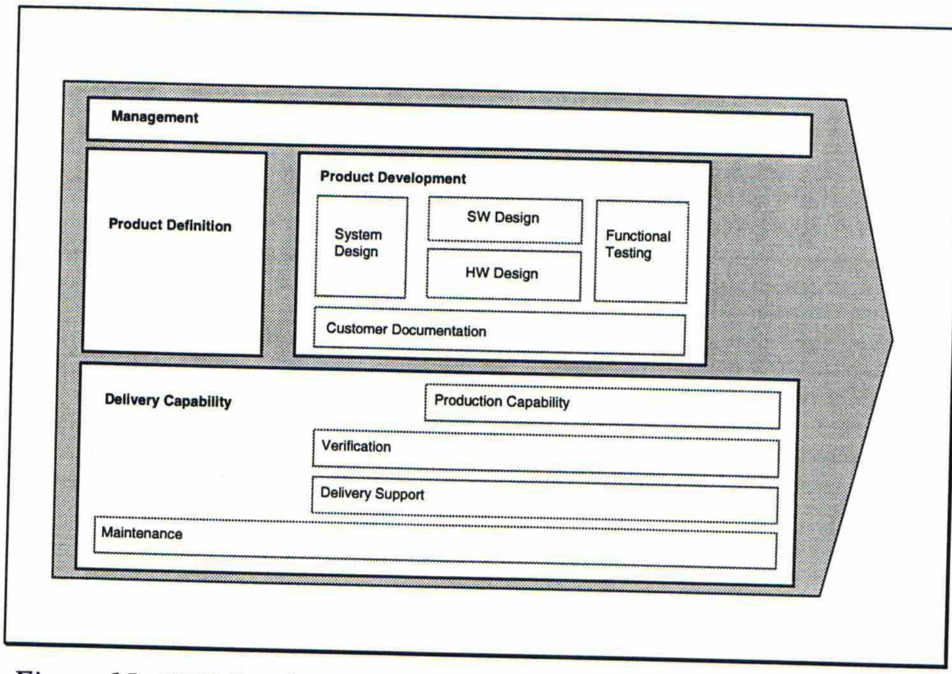


*Figure 15: SWP Product Process*

The Management Process outlines how the product development projects related to a new platform release are managed and controlled.

The Product Definition Process manages the DX 200 platform product by defining the feature contents of the new platform release and carrying out technical studies for each new feature.

The Product Development Process produces new features for the DX 200 platform product. The release commitment, made in the Product Definition Process, initiates the Product Development Process, in which the features of the new release are designed and implemented. The process is implemented in the form of R&D projects that are controlled by the Management Process. The process ends when the new features have been designed and tested, and the integration of the whole system starts. The Product Development Process is further divided into four fairly independent subprocesses: System Design, Software Design, Hardware Design and Functional Testing.

In the System Design Process, the customer requirements related to the new feature are first specified in a requirement specification document, and then the different solutions for the implementation of the feature are studied. The best implementation alternative is then selected, and the implementation is specified in an implementation specification document. Also in the process, a feature activation manual is prepared to help the customer use the new feature. The System Design Process is closely linked with the Product Definition Process, as the requirement specification documents for each new feature are actually produced in the Product Definition Process but according to the rules and procedures defined in the System Design Process. Thus, the requirement specification is, in a sense, a joint output of these two processes.

The Software Design Process designs software as specified in the System Design process. It takes the implementation and requirement specifications and uses them as the basis for the design work. While system design gives just the basic principles of how the feature will be implemented, software design produces the detailed design and implementation in the program block level. The process also tests the program modules that have been created or changed. The output of the process consists of the program code, its documents and module test material.

The Hardware Design Process designs hardware as specified in the System Design process. Based on the implementation and requirement specifications, it designs and implements the new or changed hardware product.

The Functional Testing Process tests the new or changed feature. The input to the process consists of the system design documents, software, and hardware. It then constructs the feature by combining the software and hardware, and tests the feature to see if it corresponds with the system design documents.

It should be noted that these four processes are not sequential, but they may all be happening simultaneously. This is due to the fact that each of these processes contains several activities that are performed during a long period of time (typically a couple of months).

The integration and the testing of the whole system are carried out in the Delivery Capability Process which ensures the deliverability and quality of the new release.

Figure 16 presents the relationships between the four subprocesses.

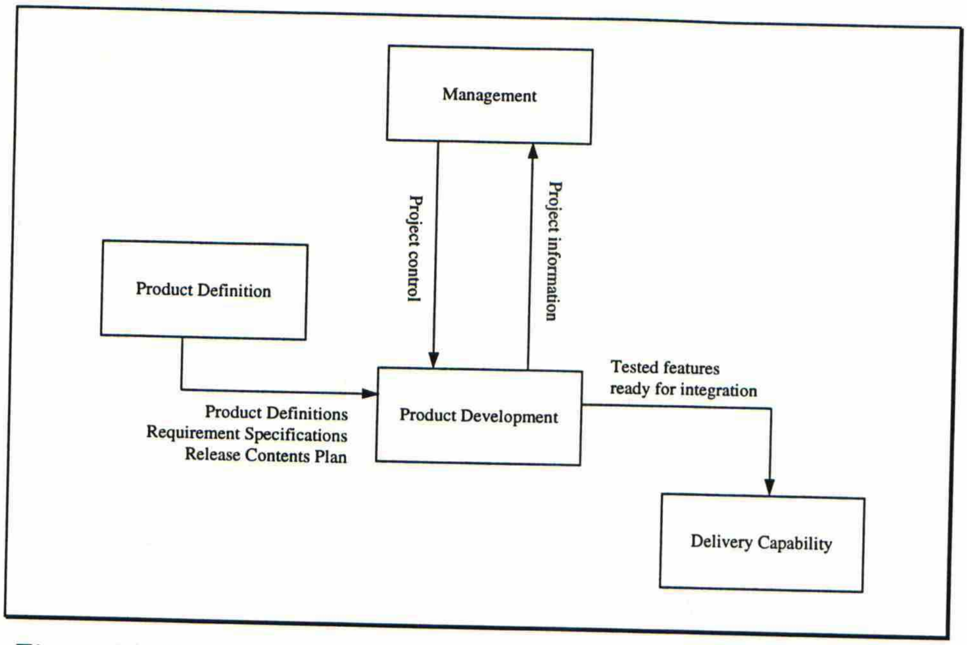


*Figure 16: The subprocesses of the Product Process*

### 3.2.3 Product Development Projects

The Product Process is a generic model of the product development activities. In practice, all product development activities in SWP/R&D are organized as product development programs. The programs are divided into projects and these further into subprojects. A product program produces a new platform release. The programs are managed with the help of the standard, phased Milestone Management Model that links the different subprocesses of the Product Process into a phased product development model.

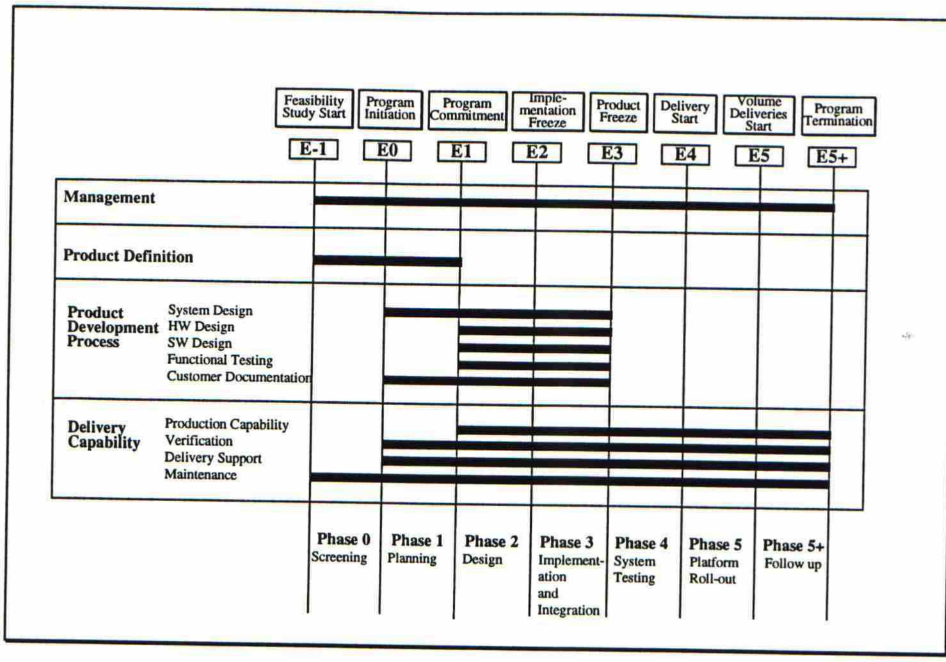


*Figure 17: The Milestone Model of the SWP Product Process*

The E-1 milestone denotes the start of a feasibility study. In E0, the program is initiated, and in E1, the R&D makes a final commitment on the feature content and schedule of the release program. After E1, the actual development starts.

E2 is a checkpoint used to control the development activities of the individual features, and E3 denotes the end of feature development and release integration (i.e. features are integrated into a testable whole). The integrated release is then system tested, and in E4 system tests are completed and deliveries start. The subprocesses and milestones relate to one another as presented in Figure 17.

The product development project organizations exist only while the project is under way. There are several projects going on all the time and developers may participate in several projects simultaneously.

As to the implementation of the process in product development projects, there are roughly two different methods in use. In one of the software departments, it is customary for designers to perform several tasks themselves. For example, one designer may design the whole feature from system design to software design and further to functional testing. In another department, the roles are more differentiated: one designer does the system design, another the software design and a third the functional testing.

### 3.2.4 Process improvement organization

Process improvement organization in SWP/R&D consists of process owners that have been nominated for the most important processes, process teams that assist the process owner in his/her work, and a coordination department that manages the process development project portfolio and coordinates process improvement throughout the R&D Unit.

The Product Process subprocess owners form a process improvement team that meets regularly to discuss Product Process improvement issues. This team has the operative responsibility for Product Process improvement within R&D.

This concludes our overview of the SWP/R&D. We will now start our discussion about the utilization of CMM in the case organization. In the next chapter, the process of using the CMM as well as the case organization experiences about the CMM are described. In Chapter 5, the usability of the CMM in the case organization is assessed based on the theoretical framework of this study.

# 4 CMM in the case organization

## 4.1 Background

The Switching Platforms R&D Unit started using the CMM as a framework for software process improvement in the beginning of 1996. There were several reasons behind this decision.

First of all, process improvement issues in general had become a point of interest in the R&D Unit as well as the rest of Nokia. Even though the company was in good shape and financial results were rising, there was a wide consensus that Nokia would have a lot to gain from process improvement. Therefore, management was ready to allocate resources to process improvement in order to further strengthen the company.

Secondly, despite all the success, software development in the company was not in as good a shape as it should have been: the problems that are familiar to almost every software company in the world were not unfamiliar to the SWP/R&D. Development projects suffered from schedule slips, the amount of work grew constantly during the projects so that initial work estimates were constantly exceeded, and the projects were unable to discover all faults from the code before delivery.

Thirdly, the CMM had by 1996 become the main software process improvement model in the world. It was largely applied and its users reported extremely promising results. Also, due to its widespread use, the CMM provided a means of comparing the present status of the SWP/R&D to other software organizations. As the process improvement team members in the SWP/R&D had agreed that using a well-established process improvement framework would be beneficial, the CMM was a logical choice.

Thus, the SWP/R&D had the means, the motive and the opportunity; the means, as the management supported SPI; the motive, since there were continuously problems in the software development projects; and the opportunity, as a suitable framework for SPI, i.e. the CMM, existed.

The utilization of the CMM started in spring 1996 by a CMM process assessment, in which the software process of the SWP/R&D was compared to the CMM model. The results of this assessment showed that the organization did not quite fulfill the requirements of the CMM level 2. Thus, a process improvement project called "CMM2" was launched to improve the software process of the organization. Further CMM assessments were carried out in December 1996 and May 1997 in order to assess how the CMM2 – as well as some other process improvement projects that were being carried out – succeeded in improving the software process as regards the CMM requirements.

As regards my role in all this, I participated in the process assessments as a member of the assessment team and in the CMM2 project as the project manager. My main role has been to provide CMM knowledge to the organization, to act as an interpreter between CMM and Nokia terminology, and to collect information about the usability of the CMM in the organization – information that is summarized in this study. The process improvement work itself has been carried out by the people working in the software process, and in this regard, I have remained an outside observer.

We will first discuss the three assessments and their results (Section 4.2) and then take a look at the CMM2 project (Section 4.3). After that, we summarize the experiences the organization had of using the CMM in software process improvement (Section 4.4).

## 4.2 CMM Assessments

### 4.2.1 First Assessment, Spring 1996

The utilization of the CMM in SWP/R&D started with a CMM-based software process assessment. The purpose of the assessment was to compare the software process of the SWP/R&D to the CMM requirements, to identify the strengths of the process as well as the areas that needed improvement, and to give a rating of the process in terms of the CMM levels. The assessment was realized with a three-person team with two participants from Nokia Research Center and myself as a representative of the SWP/R&D.

The main objective of the assessment was to get a detailed overview of the current SW process status to be used as one input for the SW strategy process. It was considered necessary to get the status description in a form that would enable comparisons with other software organizations, and this was one of the main reasons for choosing the CMM as the framework for the assessment. The assessment results were also to be used as a basis for process improvement. They would be used to support the planning of process improvement projects for 1997. Finally, the assessment results were to be used together with future assessments to follow the effects of process enhancements.

It was decided that in the assessment, only the CMM levels 2 and 3 would be addressed, as these were expected to be the most relevant levels for SWP/R&D. Remember that the level 1 is the basic level, and all organizations are, by default, at level 1. Levels 4 and 5, on the other hand, are very advanced, and it was to be expected that the SWP/R&D still had some unresolved issues on levels 2 and 3. Thus, levels 4 and 5 were considered to be not yet relevant to the organization.

After initial planning, the actual assessment process started with a management briefing in the beginning of February. The participants were the department managers of the software departments and the Head of R&D. In this briefing, the proposed approach and methods were approved.

The chosen assessment method consisted of a survey based on the CMM maturity questionnaire (see Section 2.4.4), several interviews among people working in the software process and a review of the process documentation of SWP/R&D. The questionnaire to be used was a tailored version of the SEI Software Process Maturity Questionnaire (CMU/SEI-94-SR-07), tailoring meaning that the terminology was changed to conform to Nokia terminology. Also, only the questions concerning CMM levels 2 and 3 were included in the tailored questionnaire.

The CMM maturity questionnaire is aimed at providing initial data in a CMM assessment. It contains a structured set of questions related to specific process capabilities.[78] The tailored version used in SWP/R&D contains 6 to 8 questions per each Key Process Area, and of these, there is one question for each Goal of the KPA and one question for all Common Features except Activities Performed. For example, the Requirements Management KPA has two goals, so the questionnaire has two goal-related questions and one question for Commitment to Perform, Ability to Perform, Measurement and Analysis, and Verification of Implementation, altogether six questions. Activities Performed Common Feature is not included in the questions, as answering the goal-related questions covers the activities automatically – you cannot achieve a goal without doing things right.

The assessment participants were chosen from the two DX 200 software development departments. The third software department developing methods and tools for DX 200 development was not yet included in the assessment. The group of participants consisted of line organization managers, project managers and practitioners. Altogether 54 people participated in the assessment. All of them were given the questionnaire, and 26 of them were also interviewed.

The distribution of the interviewees is presented in Table 6.

*Table 6: Interviewees of the first assessment*

| **Type of interviewee** | **Number of interviewees** |
|---|---|
| senior management | 2 |
| middle management | 6 |
| project management | 6 |
| practitioners | 8 |
| other (non-SW) departments | 4 |

All participants were provided with a one day training session in order to give them the basic knowledge of the CMM and of the assessment process before the

[78] ISPI: Enriched CBA IPI Assessment Training, 1997

actual assessment.

The participants were given two weeks to fill in and return the questionnaire. 74% of all participants, as well as 100% of the interviewees returned the questionnaire. The return percentage was quite high, considering that the questionnaire was in English, and, despite the tailoring, the language proved to be somewhat difficult to understand.

The interviews were organized not so much to check the information from the questionnaires but to get a deeper understanding of the situation experienced by the interviewees. It has to be remembered that this was an internal process assessment, not a capability evaluation performed by some outsider. Therefore, the idea was to give the participants a chance to describe the strengths and weaknesses of the software process as they see it and to get input for process improvement work, not to find hard evidence to back up the questionnaires, or to get an official maturity rating. The interviews were conducted as free form discussions about issues based on the CMM.

The document review was carried out by filling in the maturity questionnaire based on information found in the standard operating procedures of the SWP/R&D. Documentation for individual projects was not studied. The result from the document review was later compared to the results from the questionnaires the participants filled in.

The first assessment showed that the strongest Key Process Areas for SWP/R&D in CMM levels 2 and 3 were Software Process Engineering, Software Configuration Management, Training Program, Software Project Planning, and Software Project Tracking and Oversight. These results were not very surprising, as software engineering and configuration management have strong traditions in Nokia, training has been a focus of interest due to strong growth of the company, and project planning and tracking issues had, at the time of the assessment, been the object of a major improvement program in SWP/R&D.

On the other hand, the weakest KPA was Software Quality Assurance. This was not very surprising either. The traditional way of doing work in Northern Europe seems to differ from the American way in that people are given a task and then expected to complete that task without constant supervising. The underlying idea of the SQA process area in the CMM is that projects need independent, objective quality assurance which oversees that things are done properly, as planned. Especially at the beginning, this kind of thinking seemed very strange for people in SWP/R&D, thus, the rating of the Key Process Area turned out to be quite low.

Another way of calculating results is by comparing the CMM Common Features throughout the different Key Process Areas. Of the Common Features, Commitment to Perform, Ability to Perform and Activities Performed were the strongest ones in all KPAs, while Measurement and Analysis as well as Verifying Implementation were somewhat weaker. This demonstrates that the organization clearly emphasizes the activities – measuring, controlling and verifying afterwards what really was realized are not given as much importance.

When comparing the results of the two departments, the differences were quite small. This is natural, as the two departments use the same standard operating procedures and the same software development processes most of the time.

The results from the document review showed that in general, the maturity rating based on the Standard Operating Procedures (SOPs) was slightly higher than the rating from the questionnaires the participants filled in. This may indicate that the SOPs are not always followed. There is reason to ask if the processes described in the SOPs are not practical enough to be applied, or if the employees just do not know the processes, i.e. they do not read the SOPs.

These results were presented in April to the participants of the assessment as well as the SWP Product Process Team that is responsible for steering and controlling process improvement projects. The results were largely discussed in the whole organization during the following months. As the results showed that the SWP fell a little short of level 2 in the CMM, a process improvement project was started in October '96 to improve on the "missing parts". This project is more closely presented in Section 4.3.

### 4.2.2 Second Assessment, December 1996

When the CMM model was decided to be used as the framework for software process improvement in the SWP/R&D, it was also decided to repeat the CMM assessment quite often, approximately every six months. This way, the organization would get enough measurements to assess trends in process improvement, and, also, the assessments would serve as a way of spreading information about the CMM.

The second assessment was carried out in December 1996. This time, the assessment method was a lighter one: the participants were provided with a 1.5-hour training session, and 2 hours were used for filling in the questionnaires. No interviews were conducted this time, but a document review was carried out the same way as in the first assessment. Just as in the previous assessment, only the CMM levels 2 and 3 were considered. Altogether 86 people participated in this assessment, all from the software development departments. 81 people returned the questionnaire.

Several changes were made in the questionnaire for the purpose of the second assessment. The language of the questionnaires was further tailored to suit Nokia terminology, and some complicated sentences had been rephrased, since, after the first assessment, many people complained that the questionnaire had been very tedious to fill in. Secondly, the answer scale was changed. In the first assessment, the available choices were yes/no/don't know/does not apply. This yes-no scale was considered awkward, as reality is never quite so black and white. For the second assessment, the scale was changed to fully satisfied/largely satisfied/partially satisfied/not satisfied/does not apply. The results of the first assessment were converted to this new scale in order to enable comparisons and trend calculations. Thirdly, the questions were ordered into two groups: goal-related questions and questions related to the CMM Common Features. The idea in the CMM is that the goals must be fulfilled in order to fulfill the requirements of a maturity level, and the Common Features work towards fulfilling the goals. An organization may have the same practices as presented by the Common Features, or it may have alternative practices that produce the same end results as those presented in the CMM. It was considered necessary to separate these two groups from one another so that the goals would be clearer to everyone.

Also, the way the questionnaires were filled in was changed. In the first assessment, the participants filled in the questionnaire all by themselves. This time, the participants filled in the questionnaire in one 2-hour session in which they had the possibility to ask questions or for clarification about the questionnaire.

When discussing the results of an assessment based solely on the questionnaire, one has to remember that the accuracy of the results can be low for several different reasons. For example,

- the questions were considered difficult to understand, and as it has turned out, people understood the questions in a variety of ways
- the questionnaire was originally planned to be filled in by project managers, but in SWP/R&D, the participants represented different employee groups
- the profile of the group answering the questionnaire, in terms of work position and experience, was slightly different from the first assessment, as there were more new employees and fewer managers represented; thus, comparisons between the two assessments must be made with caution.

Even SEI (Software Engineering Institute) recommends that the questionnaire be used only as a starting point for the assessment, and that it cannot be relied on for giving an accurate maturity rating for an organization. Therefore, when interpreting the results from the questionnaires, one has to be very cautious.

Remembering this, the results were nonetheless rather interesting this time. In many cases, the results now were lower than the results in the first assessment. Especially in Requirements Management and Configuration Management the rating in goal-questions was clearly lower than the previous time. The most plausible explanation is that the level of expectation of the participants had risen from that of the first assessment. The participants were given explanations about the questions so that they better understood what was being asked. Also, ever since the first assessment, the CMM related issues had been the topic of a lot of discussion, so people might have had higher standards by the second assessments. Another point to remember is that the participants were now given several answering alternatives instead of just yes and no. Therefore, they could now fine tune their answers to correspond to reality.

Actually, SEI reminds us that it is quite common for the results in an organization to drop in the second or maybe third assessment, before they start to rise. This is mainly due to the fact that the second or third time around, people have higher expectations, but the process improvements based on previous assessments have not yet been properly implemented. It seemed that this was exactly what was happening in SWP/R&D.

When comparing the departments, the differences were again quite small. One interesting point could be seen regarding the Äänekoski unit. It seemed that the rating for the Äänekoski unit was consistently a bit higher than for the departments in Helsinki area. One explanation could be that since the Äänekoski site is considerably smaller than the departments in Helsinki area, information flows more freely there and people are more aware of the process improvements that are being carried out. Also, there were relatively more project leaders in the group of assessment participants in Äänekoski than in the assessment groups in Helsinki. It is possible that project leaders have a more positive view of the software processes being used.

Of the CMM Common Features, the areas that were weaker in the first assessment had now risen so that the results for all Common Feature related questions were approximately the same. Thus, it seems that measurement and analysis as well as verification issues had developed between the two assessments.

The assessment results were again presented to the assessment participants as well as process improvement team members.

### 4.2.3 Third Assessment, May 1997

The third CMM assessment was carried out in May 1997, and again, a new method was used. This time, a group of seven to nine people was chosen from each software department. The groups each had a meeting to discuss the issues in the CMM questionnaire and based on the discussion, the group answered the questionnaire on a consensus principle. Thus, only one questionnaire was filled in for each of the two software departments. Main points from the group discussion were also recorded in order to use them in the assessment report. The idea is that, this way, we can have some further information in addition to the questionnaires, and yet save some time, as organizing e.g. interviews to collect information takes a lot of time and energy. Just as in the previous assessments, only the CMM levels 2 and 3 were considered, as there still were open issues on these levels, and thus, the levels 4 and 5 were not yet relevant to the organization.

Also, this time the third software department that develops tools and methods for the DX 200 departments, was included in the assessment. Finally, a fourth group was formed with representatives from the support department, system testing department, project coordination department and product management department. Altogether 30 people participated in the assessment.

This time, the assessment questionnaire was exactly the same as in the second assessment. A document review was not considered necessary this time, as there had not been any major changes in the SOPs between the second and third assessments. However, there are a few process improvement projects going on that will cause some significant changes in the processes and SOPs that are applied. These changes will be implemented mainly during the rest of this year. Thus, a document review will be necessary in the next assessment. More information about these projects is presented in Section 4.3.

The final results show that, first of all, there is a clear upward trend in the assessment results. For all KPAs, except for two, the rating based on goal-related questions has risen from the second assessment. The two exceptions are Software Quality Assurance and Organization Process Definition, and major process improvement projects are currently focusing on these issues. Thus, it is expected that the ratings for these KPA's will rise considerably in the next assessment to be held in January 1998.

Just as in the previous assessments, the differences between the DX 200 departments were small. However, the rating for the methods development department was slightly lower than for the DX 200 departments. This may be due to the fact that this was the first assessment for that department, so CMM-based improvements are just beginning there.

The main aspects from the three assessments are summarized in Table 7. Next, we will discuss the CMM-based process improvement projects being carried out in the SWP/R&D Unit.

*Table 7: Comparison of the Three Assessments*

| | Assessment 1 | Assessment 2 | Assessment 3 |
|---|---|---|---|
| **Time** | April '96 | December '96 | May '97 |
| **Assessment method** | Questionnaire, interviews, document review | Revised questionnaire, document review | Revised questionnaire, group discussions |
| **Number of participants** | 54 | 86 | 30 |
| **Scope of assessment** | Two DX 200 software development departments | Two DX 200 software development departments | Two DX 200 software development dep'ts, methods development dep't, support, testing, project, product management dep'ts |
| **Strongest KPA (for whole R&D)** | SW Product Engineering | SW Project Planning | SW Project Planning |
| **Weakest KPA (for whole R&D)** | SW Quality Assurance | SW Quality Assurance | SW Quality Assurance |
| **Strongest Common Feature (for whole R&D)** | Commitment to Perform | Commitment to Perform | not available |
| **Weakest Common Feature (for whole R&D)** | Verifying Implementation | Verifying Implementation | not available |

## 4.3 CMM2 Project

A process improvement project called "CMM2" was launched in October 1996. After the first CMM assessment in the spring of 1996, the management team of

R&D had set a process improvement target of fulfilling the CMM level 2 requirements by the end of 1997. As the results showed that some work had to be done to reach this target, CMM2 was started in order to follow and coordinate the progress of this work. The name of the project referred to CMM level 2, as the level 2 was the target of the project. Note that while the CMM process assessments considered both level 2 and level 3 issues, the case organization decided to focus its process improvement efforts mainly on level 2 issues before concentrating on level 3. This is natural, since level 3 can genuinely be reached only after level 2 requirements have been fulfilled. The information about level 3 issues collected in the CMM assessments was not wasted, however, as some groundwork for level 3 has been done, and some level 3 issues have already been considered when improving on level 2 issues.

The objective of the CMM2 project was to ensure that all necessary improvement initiatives have been introduced, and to follow the completion of these initiatives. In order to achieve this, the project was decided to be carried out by using 4 - 8 person task forces for each of the six Key Process Areas in CMM level 2. The task forces were to complete the following activities:

- familiarizing themselves with the Key Process Area (KPA) in question
- familiarizing themselves with the KPA results of the CMM assessments carried out in SWP
- planning the needed actions to reach CMM level 2 in the KPA during 1997
- planning the implementation of these actions
- monitoring the implementation and give a report to the CMM2 Project Manager and Steering Group.

As can be seen, the task forces were not defined to be responsible for implementing all improvements, but for ensuring that someone implements them.

I was nominated as the project manager in October 1996, and the task forces started their work in December 1996. As the task force members were not on this project full-time, it was decided to have meetings approximately twice a month, and in the mean time, each member collected further information.

The first task for the task forces was to get to know the KPA in question: what are the goals and key practices, how the CMM needs to be interpreted to suit this particular organization, and so forth. After this was completed, each group studied the assessment results in detail in order to analyze what needs to be done in that KPA to fulfill the CMM requirements. This took approximately 3 months, so in March, each group had completed their study of the CMM requirements and the assessment results.

The next step was to prepare action plans for each KPA. These plans consisted of detailed tasks, schedules and persons responsible for carrying out these tasks. The aim of these action plans was to ensure that the CMM level 2 requirements would be fulfilled. At this point, the Subcontract Management group terminated its work, as the SWP/R&D had used outside subcontracting only on rare occasions, which could not be considered as part of normal activities. The group concluded that at that point, it had clarified the CMM requirements and the current status of SWP/R&D as regards those requirements. In case subcontracting in the future becomes more common, the results of the group's work could be used to improving subcontracting related capabilities.

The other five groups continued their work and prepared action plans. These plans were reviewed and approved in May 1997. These plans are, at the moment, being carried out. We shall take a closer look at each of the task forces and discuss the contents of the plans the task forces formulated.

### 4.3.1 Requirements Management Task Force

The purpose of the CMM Requirements Management process area is to establish a common understanding between the customer and the software project of the customer's requirements that will be addressed by the software project.[79]

[79] CMU/SEI-93-TR25. p. L2-1

For the requirements management team, one of the first problems was "translating" the CMM language into terms more familiar in Nokia. The CMM uses many generic terms like: allocated requirements; system requirements; project; software manager; managed and controlled documents; and so on. Each of these terms had to be mapped to this specific organization before being able to say if the requirements would be fulfilled or not.

After reviewing the CMM requirements and the assessment results, the task force concluded that SWP/R&D actually has a lot of good practices, but apparently the software designers, and in some cases, even project managers, are not exactly aware of them. The problem is that in SWP/R&D, the development project organization is usually established after the requirements have been specified. Therefore, the project organization, including both the project manager and the designers, does not have a clear view of the Product Definition Process, which has a central role in requirements specification and management.

The focus of the action plan was, therefore, in enhancing the flow of information between the Product Management department and the development projects. Also, the idea of establishing the project organization at an earlier stage was promoted in the team. The idea was that, if the development project manager is nominated at an early enough stage, (s)he will have the opportunity to follow the requirement specification and thus have a broader view of what the project is supposed to achieve and why.

### 4.3.2 Project Planning and Project Tracking and Oversight Task Forces

The purpose of Software Project Planning is to establish reasonable plans for performing the software engineering and for managing the software project,[80] while the purpose of Software Project Tracking and Oversight is to provide adequate visibility into actual progress, so that management can take effective actions when the software project's performance deviates significantly from the

[80] CMU/SEI-93-TR25. p. L2-11

software plans.[81]

The Project Planning and Project Tracking and Oversight task forces decided to combine their action plan into one, as these two process areas are very closely linked together.

These two task forces had a somewhat easier job than the requirements management team, since the terminology was much more suitable for Nokia without any tailoring. Some discussion was caused by e.g. the requirement to estimate the size of the work product to be developed in the project planning phase. Since SWP/R&D is developing and maintaining an existing, large system, all lines-of-code based size measurements are unsuitable for use. Thus, finding a satisfactory measurement for size posed some problems.

The task force concluded that the project planning and tracking issues were in good shape, and as the assessment results prove, these areas were already quite close to fulfilling the CMM level 2 requirements. Some further improvements would be introduced with the implementation of a new project management tool, and the project management manual which is due to be released in December. The project planning and tracking action plan focused mainly on listing some issues that would have to be included in the manual or taken into consideration when further developing the project management tool.

### 4.3.3 Software Quality Assurance Task Force

The purpose of Software Quality Assurance is to provide management with appropriate visibility into the process being used by the software project, and of the products being built.[82]

The software quality assurance task force had much bigger problems to solve than those mentioned earlier. As discussed in Section 4.2, SQA was consistently the weakest of all level 2 and 3 process areas. Part of the reason was that in SWP/R&D, quality assurance had been organized on a line organization basis,

[81] CMU/SEI-93-TR25. p. L2-29

[82] CMU/SEI-93-TR25. p. L2-59

thus, each department had a quality manager. Since all product development work is done in projects, the department quality managers would naturally deal with project quality issues as well as line organization quality issues. However, project as such were not an object of quality assurance. However, the CMM requires that an outside, independent quality assurance is organized for all projects.

At first, the CMM way of thinking raised a lot of objections: project quality assurance as presented in the CMM was seen as a "police organization" that would watch over the projects. In an organization where people take pride in fulfilling their tasks without someone watching over the shoulder, this was seen as a very uninteresting option. However, when the CMM was further studied, the reasoning behind it started to become clear. When a project manager is responsible for making sure that the project is carried out on time and within budget, it is too much to ask that (s)he also acts as an objective quality assurance person in the project. Thus, an outsider could be seen as supporting the project: providing the project group information about quality and process related issues and giving an outsider's view of how the project really is progressing. One software organization after the other has seen that without an outsider's perspective, people tend to make too low work amount estimates in the beginning and then are too optimistic about being able to complete the task in time. Software projects, in which the work amount grows 100% from the original estimate and the schedule is delayed months or even years, are just too common.

Thus, after a lot of discussion, the task force decided that it would be worth it to try establishing a project quality organization. It was decided that each software department would choose a few people to act as either part-time or full-time project quality assurance persons (PQPs). These people were nominated in May 1997. At the same time, the work description of a PQP was being prepared. The PQPs role was defined to be particularly active in the project planning phase, as (s)he would ensure that the plan is prepared in accordance to the processes and procedures of the R&D Unit, that all estimates in the plan are realistic and that the people participating in the project are committed to the plan. During the project, the PQP would follow the progress from both a process and a product point of view by checking review minutes and by participating in some reviews. In case the PQP notices uncontrolled deviations from the project plan, his/her responsibility is to discuss this with the project manager and group. If no solution can be found, the issues are escalated to higher levels in the organization. Uncontrolled changes mean changes that have not been recorded anywhere, that have not been approved by appropriate authorities and the risks of which have not been estimated.

So far, the PQPs have been given training in quality and project issues as well as the CMM. At the moment, the first projects with a PQP have been started, so none of them have much experience yet. However, the first impressions seem quite positive; the project groups have accepted the PQPs well and the PQPs as quality professionals apparently can provide a lot of help to the project groups.

### 4.3.4 Software Configuration Management Task Force

The purpose of Software Configuration Management is to establish and maintain the integrity of the products of the software project throughout the project's software life cycle. The activities in SCM involve identifying the configuration (i.e. naming of the components, version numbering, etc.), systematically controlling changes to the configuration and maintaining integrity and traceability of the configuration throughout the software life cycle[83] (e.g. rules of approval for changes, audits and reviews, reporting about the changes, building of the software products from their components, etc.).

The Software Configuration Management task force had quite a lot of work to do. Configuration management was one of the strong process areas in the assessment, but it seemed that while some areas in configuration management, like version management, were well established and functioning, there were problems in change management and in the rules of approval of the products.

On further complication to the work of the task force was that in SWP/R&D, configuration management had not been seen as a process itself, but instead it

[83] CMU/SEI-93-TR25. p. L2-71

was spread out into many small pieces. Thus, there were a lot of process descriptions and standard operating procedures that each contained a small part of configuration management, but no document described configuration management as a whole.

The task force started its work by defining configuration management: what configuration management was, what activities belonged to it and what kind of documentation did SWP/R&D have concerning these activities. When this groundwork was ready, the team went through the CMM requirements and the assessment results and then formulated the plan.

The plan focused on preparing a high level description of configuration management that would act as "an umbrella" for all other documents containing SCM related issues. This description is currently being prepared. It will contain generic descriptions of the product baselines that are created during the development of a new release and the management of these baselines including identification, change management, approval and so on. It will also contain references to the documents in which these activities are presented in more detail.

## 4.4 Experiences from the use of the CMM

We have now covered the CMM assessments and the CMM2-process improvement project. Now it is time to turn our attention to the experiences the case organization has of using the CMM: a lot has been done with the CMM, but what kinds of benefits have been received?

There are several sources of information for collecting experiences about the CMM. First of all, the assessment participants have been asked for their opinions about the CMM, the assessment process and the changes they have seen happening due to the CMM. Also, the task forces have been a valuable source of information, as a lot of information has been available in the task force meetings. To back up these verbal sources of information, a survey was conducted in which the task force members were asked to rate their opinion about the CMM and its capability to support process improvement. We shall now discuss each of these sources of information in turn.

### 4.4.1 Experiences from the Assessments

The assessment participants in the CMM assessments have been asked both to comment the assessment process being used and to present their opinion about the usefulness of using the CMM in process improvement.

The most frequent comment has been that the CMM assessments are relatively hard work for the participants. Even though the questionnaire has been tailored, there still are things that are hard to understand or to map to the organization. Part of the problem is that the assessment participants are not used to thinking in terms of processes – their everyday work revolves around products and technologies.

Another comment, especially in the second and third assessment, has been that the assessments just dwell on the same old problems and yet nothing changes in the way work is done. This is interesting, because when comparing the assessment reports and the questions that have risen in each assessment, some major changes can be seen. Many of the problems in the first assessment have now been solved and yet new ones have appeared. This gives reason to ask if the designers do not get enough information about process improvement work or if the expectations in the organization are unrealistic. Either way, this is a serious problem as it directly affects the motivation the designers have towards process improvement. If process improvement is seen as a series of projects that never do anything really good and useful, people soon stop participating in it. An attitude of "maybe it will go away if we are really quiet" easily gains ground.

The assessment participants have also expressed doubts about the reliability of the assessment. They reason that since it is so hard to understand some questions, the answers are more or less random and thus not reliable. Part of this concern is quite justified, since as has been mentioned, the questionnaire based method cannot be considered very reliable. However, the results of the assessment seem to be consistent enough and also accepted by the organization to a degree that indicates that with proper caution, the assessment results can be used to give indication about the status of the software process.

As far as the way the assessments are carried out is concerned, each method seemed to have its advantages and disadvantages. The first assessment, with its interviews, gave a lot of information yet demanded a lot of work. Also, people considered filling in the questionnaire without any outside help very tedious. The second assessment was easier on the participants: they were given assistance in filling in the questionnaire, and the assessment did not take a lot of time. However, the amount of information collected was low. The third assessment seemed to be a compromise between the two previous ones. It did not take as much time as the first assessment, and yet the group discussions provided information in addition to the questionnaires. Also, variations in results caused by individual interpretations of the questions were reduced as the questionnaire was filled in with a group consensus.

However, the consensus may lead to "choosing the middle road", i.e. choosing the alternatives in the middle (largely/partially satisfied) instead of fully/not satisfied in order to make everyone happy. In some cases, one member of the group considered something fully satisfied and another not satisfied. When the group then chooses largely or partially satisfied, information is lost as the variation is not shown in the questionnaire.

Another problem related to the third assessment method is that some participants felt the assessors may have influenced the assessment so that the assessment result ended up to be higher than what the assessment participants thought realistic. A counter-argument is that in many cases, it was evident that people found it very difficult to say something is fully satisfied. For example, the CMM requires that a plan is prepared for all projects, and if this is done in the organization, the answer should be fully satisfied even if the plan does not cover all the things it should, or the responsibilities for its preparation are not always clear. The problem is that the contents of the plan and the responsibilities are addressed in *other* questions, and thus should not be allowed to influence the first answer. Otherwise there is a risk of answering something else than what was actually asked. Another factor that systematically tends to raise the rating is that while all goals are included, the questionnaire contains only a sample of the common practices. Thus, in the discussion the issues that are not included in the questionnaire are not taken into account when answering the questions. Thus, even if some of the practices not covered by the questionnaire are not well handled by the organization, a questionnaire based assessment totally ignores this.

One further comment concerned the scope of the assessment. In SWP/R&D, the assessments have been done on a department basis: each department has been given a rating. However, there are several projects within each department, and the projects have various different ways of doing things. Thus, getting a rating for a department may prove almost impossible as different projects within the department have opposing practices. This is partially due to the fact that the CMM has been planned from a project point of view. Even at the risk of losing the possibility of calculating long time trends, maybe it would be more advantageous to carry out further assessments on a project scope.

Altogether, the assessment participants have had a positive attitude towards CMM based process improvement and despite the critique, most seem to think it is a good idea. The participants have also commented that it is good that people are given a possibility to voice their opinions and to give input towards process improvement work.

### 4.4.2 Experiences from the Task Forces

The task forces have been a valuable source of information, as the participants are experts in the respective process area and they have had a chance to really get to know and use the CMM in their area of expertise.

The first problem with the CMM was with the language and the need to tailor the model for the organization, which took a lot of time and energy. It also caused problems like those experienced with the quality assurance process area. When the CMM speaks in very generic terms, it is easy to misunderstand and thus miss the point of the process area totally. This same phenomenon has been witnessed in many occasions also outside SWP/R&D. People read the CMM carelessly and without clarifying the meaning of each term. This leads to misunderstandings, that are discussed e.g. by Wiegers[84] in his article.

Another problem is that the CMM only addresses the key process areas. While this provides a way to prioritize improvement projects, it also makes it impossible for an organization to rely on the CMM alone. Many important issues (e.g. human resource management) are not covered by the CMM, so some other model is needed to help in the development of these issues.

On the positive side, the task force members consider that the CMM gives a solid basis for process improvement. It addresses the issues in a logical, thorough way. Also, despite the sometimes complicated use of the English language, it still is a very practical model – it clearly addresses the issues that are important for any software organization. Actually, SWP/R&D has already applied the CMM also to documentation processes and some parts of it will be applied to hardware development as well.

It seems that many task force members have become "CMM-believers" in that they actively use the model as a supporting framework in process improvement instead of just seeing it as a list of requirements to be fulfilled.

### 4.4.3 Information from Survey Results

A survey was conducted among the task force participants to further clarify their opinions about using the CMM. The task force participants were chosen as the respondents for two reasons. First, they had familiarized themselves with the CMM, so they had the necessary understanding of the model. Second, they had been participating in CMM based process improvement for a little over 6 months, so they had first-hand knowledge about the actual use and usefulness of the CMM in SWP/R&D.

The survey had 6 questions, each with a five-point answer scale as follows: 1) None, 2) Weak, 3) Average, 4) Strong, 5) Excellent. The survey was sent to 28 people and 18 of them returned the survey, thus, the returning percentage was 64.

---

[84] Wiegers, Karl. 1996

The questions in the survey were related to the use of the CMM in each of the phases of the Plan-Do-Check-Act cycle. Based on the conclusions presented in the literature study about the usefulness of the CMM in process improvement, the hypothesis was that the questions related to identifying improvement opportunities, choosing the projects and assessing the results, i.e. questions 1,2 and 5 would get a higher rating than the questions related to the detailed improvement planning and implementation (3 and 4). Question 6 was included for getting an overall rating of the usefulness of the CMM.

The questions of the survey and the results are presented in Table 8 and Figure 18.

*Table 8: The survey results*

| **In your opinion and based on your experience, what kind of support does the CMM provide in the following:** | | **Score** |
|---|---|---|
| 1. | Identifying software process improvement (SPI) opportunities | 4,11 |
| 2. | Choosing and prioritizing the SPI opportunities to be implemented | 3,06 |
| 3. | Planning the SPI projects in detail (what, who, when) | 2,83 |
| 4. | Implementing the SPI projects | 2,67 |
| 5. | Assessing the results of the SPI projects | 3,56 |
| 6. | How would you assess the overall usefulness of the CMM in process improvement work in SWP/R&D? | 3,72 |

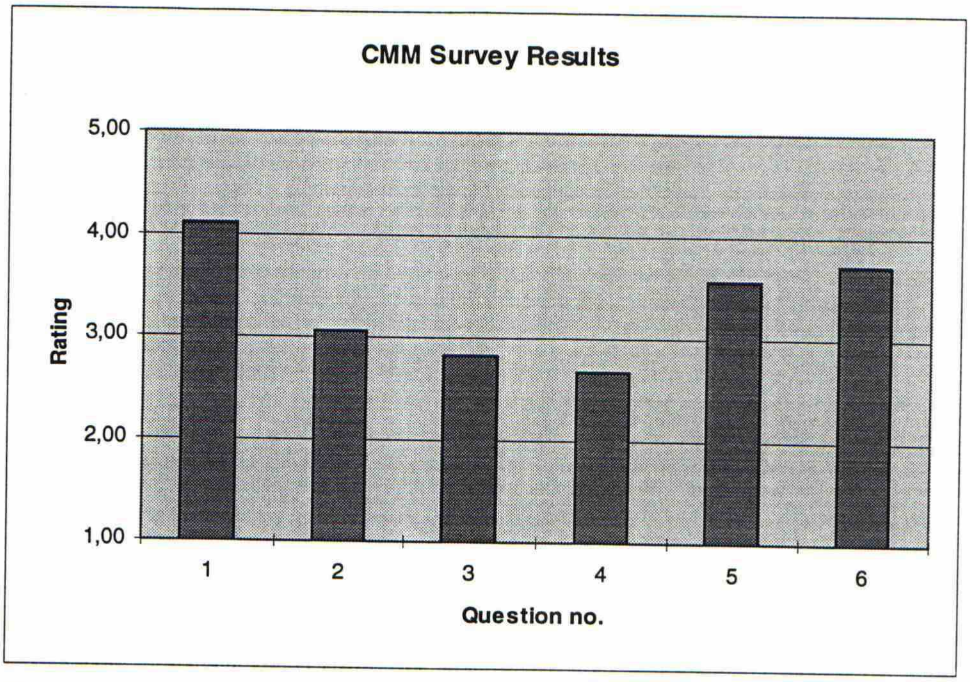


*Figure 18: Survey results as a graph*

It seems that the hypothesis made before the survey realized quite well. The ratings for questions 1 and 5 are clearly higher than for questions 3 and 4. Another interesting point to note is the relatively high rating given for question 6, that was about the overall usefulness of the CMM. It seems that at least the people who have participated in CMM-based process improvement themselves, have a very positive attitude towards the model.

One question that rises from the results, is the relatively low rating for question 2. According to the hypothesis made before the survey, the CMM would provide significant help in choosing and prioritizing process improvement initiatives. This does not seem to be the case in the case organization. Considering the fact that the case organization has a relatively large number of process improvement projects going on, and that these projects have many different scopes in addition to the CMM, there is reason to ask if the organization should focus its process improvement resources more clearly on a few well chosen areas such as those suggested by the CMM.

In addition to the questions, the survey respondents were asked to present any further comments they might have. Only some respondents presented any comments, and these were:

- "CMM work is proceeding well in SWP and the method offers a good frame for explaining the importance of the improvements"
- "We can also identify the problems without the CMM, and implementation has never been our strong side. It seems that improvements are being carried out with too little resources"
- "Small department-based assessment groups will probably give a different rating for the department each time, as the participants' backgrounds and knowledge about the department varies strongly, as well as the time they have been employed. It is difficult to consider the result very significant from this basis."
- "The CMM can easily lead to exaggeration, because in standard operating procedures even small details may seem very important. Everything should not be realized without critique."
- "I believe in the CMM-model as these issues have been found to be good and important in other companies."

Thus, the verbal comments included some critique towards both the model and especially the way it had been applied in SWP/R&D.

# 5 Assessment of the Usability of the CMM

We have now discussed the history of using the CMM in SWP/R&D, and also presented the experiences gained in using the model. Now it is time to assess the suitability of the CMM for software process improvement in the case organization and, thus, to create a basis for an Action Plan.

The experiences gained from using the CMM in the case organization demonstrate clearly the importance of tailoring and interpreting the model when using it. The case organization has had to tailor the questionnaire as well as the way the CMM assessments are carried out, in order to enhance the usability of the model. Also, the CMM requirements had to be interpreted in the CMM2-project in order to be able to use the CMM as a basis for further improvement. The work amount needed for the tailoring should not be underestimated, as this is a considerable task for any organization using the CMM.

To further analyze the usability of the CMM in the case organization, we now return to the theoretical framework of this thesis and discuss the relationship of the CMM to each component of the PDSA-cycle as well as consider the role of the CMM as a supporter or a promoter of process improvement.

## 5.1 PLAN: CMM and Process Improvement Planning

In the case organization, the CMM's role has been quite remarkable in planning new process improvement projects. The CMM2-project is the biggest process improvement project in the process improvement project portfolio this year, and a new "CMM3-project" that will presumably be initiated to follow the CMM2-project, will probably take a significant proportion of process improvement resources next year.

The usability of the CMM in process improvement is a two-sided question. On one hand, the CMM does provide a way for identifying process improvement opportunities and for prioritizing the realization of these opportunities. However, as one survey respondent (see 4.4.3) as well as some participants of the CMM assessments commented, we have been able to identify and prioritize opportunities even without the CMM, and that the model has not really told us anything we did not already know. As a counter-argument it could be said that the CMM does in any case provide a structured approach that is based on the experiences of many major software organizations, and that forces us to deal with *all* problems discovered while using the model. Without a framework, it would be very easy to "forget" some awkward problems instead of trying to solve them.

Another problem with the CMM is that it does not provide support for the detailed planning of the improvement projects. Its approach that concentrates on the 'What' and not the 'How' may leave an organization hanging in the air; we know what should be done but the model does not help in how to achieve it. Naturally, the reasoning is that an organization must be hard-working enough to define the best way to implement the changes itself. This is however sometimes very discouraging. The case organization of this thesis found out that, for instance, some examples of successful solutions could inspire the creativity in the organization, so that the implementation of improvements could be started faster.

A third problem with the CMM in planning improvements is that there really is no hard evidence on why the areas presented in the model should have any priority in process improvement. As presented earlier, some studies exist about the benefits of using the model, but as the CMM is based more on experience than on a solid theoretical foundation, there are no guarantees of success. Thus, claiming that the improvements required by the CMM should be the first ones to be implemented is more a declaration of a firm belief in the model than a presentation of hard facts. The CMM does not bring any fast results and thus may be difficult to justify for people who need to do something about e.g. cycle time right now. On the other hand, those "fast results" are often more a temporary relief than a permanent improvement, when the underlying problems are ignored. The CMM aims at building a solid project and process management basis, not to rely on quick fixes.

## 5.2 DO: CMM in Implementing Improvements

As was discussed in the literature study, the CMM provides very little support for actually executing the process improvement projects. This has been noticed very clearly in the case organization, and in some occasions, this has caused the model to be interpreted in a wrong way. The wrong way means that when the CMM as such does not provide detailed instructions, people start to read it very literally in order to find the needed detailed instructions "between the lines". On many occasions, people have said things like "CMM says that we must create a software quality assurance group" when actually it says that someone has to be assigned the responsibility of taking care of the SQA activities, but not that a new organizational unit has to be created for it. Thus, the high level of abstraction of the CMM may cause problems when it is taken too literally.

Another problem is that some people seem to become frustrated with the level of abstraction of the model. Thinking in terms of processes is not very easy for product development people who have become accustomed to thinking in terms of products and organizations. Giving these people a high-level process model like the CMM poses problems if they are not provided with the appropriate training in process issues. Thus, using the CMM in an organization clearly calls for the training of everyone working with the software process.

## 5.3 STUDY: CMM in Studying Results

The CMM assessments are a natural way of measuring and studying the way process improvement progresses. However, they are not without problems, either.

In the case organization it has become clear that the assessments have to be tailored to the needs of the organization. As there are several ways of conducting assessments, with varying reliability and cost, the organization must weigh the pros and cons of each method and find the way most suitable for itself. In SWP/R&D, this search for an assessment method is still going on, as the future assessments will, for example, try the project approach instead of the department approach. The choice of an assessment method is important for many reasons, the smallest of which is not the trust the people working in the organization have towards the chosen method. In the case organization, the assessment methods have played an important part when discussing the reliability of the results. The questionnaire-based methods have been seen as pretty unreliable, as the questions were considered hard to understand and answer.

Another point to remember is that the CMM assessments do not provide much information about individual process improvement projects or even process improvement in general, for the short term that is. The issues the CMM deals with usually take a long time to change, and moving from one maturity level to another is almost always a question of one or more years. Therefore, the CMM in itself is not an adequate measurement for process improvement. An organization must have other, faster ways of showing that things are really progressing. People tend to wait for visible changes in a period of a couple of months, and if nothing seems to happen, motivation soon dies. The case organization of this thesis has strongly developed the management of process improvement projects in the last few years, but there is still room for improvement.

## 5.4 ACT: CMM in Continuous Improvement

For continuous improvement, the CMM again has little to offer. After the CMM assessments are carried out, the organization is on its own. It has to define, how the deviation in results and plans is to be corrected and how the next maturity level is to be reached. The CMM is good in that it always gives an organization something to reach for, as even a level 5 organization can always improve its processes. Thus, the CMM's main role in this regard, is that it provides the "shining star" that the organization can aim for in process improvement. There is, however, reason to ask if the CMM is enough. Maybe in addition to the CMM, an organization would find it beneficial to use some other model, for example SPICE, that provides more information about the "how"?

## 5.5 CMM as a Supporter or a Promoter

In the case organization, the CMM acts both as a supporter and a promoter. It supports the process improvement, in that the repeated process assessments prevent the organization from letting some earlier improvements from slipping back to where they started. This has been demonstrated, for example, with project management. When the CMM project was started, project management had already been a point of interest in process improvement for quite a while. Without the CMM, it is very probable that other issues would have been raised to the foreground, but the CMM forced the continuous improvement of project management practices.

On the other hand, the CMM is a promoter since it has given all software process improvement projects a common aim and thus automatically coordinates process improvement. This is an important point, since, in a large organization, it would be easy to have competing process improvement projects that have conflicting objectives. The CMM gently forces everyone to work in the same direction.

## 5.6 Conclusions: CMM's role in the SWP/R&D

As a summary to this discussion, we return to the theoretical framework of this thesis that was presented in Figure 11. Based on the preceding discussion, the figure is now updated to describe the SWP/R&D point of view to the CMM in software process improvement. This updated picture is presented in Figure 19.

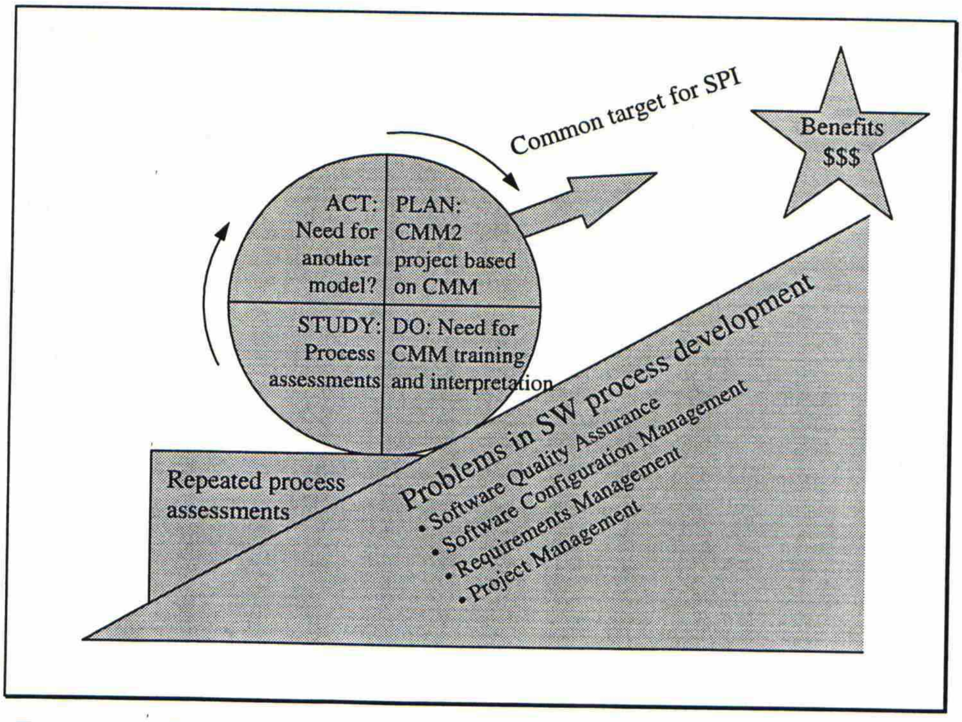


*Figure 19: CMM in the case organization*

We can now conclude, that while the CMM has definitely provided a solid basis for software process improvement in the case organization, it is by no means a "silver bullet": it will not solve any problems automatically. Using it requires commitment, training and a long term plan. This is what we will discuss in the next chapter.

# 6 Conclusions: Action Plan for the Case Company

In this chapter, an action plan is presented for the case company, the main results of the study are summarized and some further research areas are suggested. Before presenting an action plan for the case company, it is, however, necessary to assess the validity and reliability of this study.

The theoretical framework of the thesis was built based on well-established models presented in the literature, and the application of the framework in the case organization produced an understanding that seems to be consistent with the way the case organization employees see the situation themselves. Therefore, as the foundation of the model is quite solid, and its application in the case organization produced reliable results, the validity of the model can be assessed to be relatively high. Also, the framework is of such generic nature that there does not seem to be any obstacles for applying it to other organizations, as well.

The case study is, to a large degree, based on discussion and interviews with the case organization employees. Thus, there is a risk of subjectivity in the data collected. However, the amount of people who participated in the CMM assessments as well as the CMM2-project was quite high, and the results presented in this thesis have been extensively discussed in the organization. Based on the discussions, the results of the study seem to correspond well the reality of the organization. Also, the results of the survey support the results of this study. Thus, the results of the study can be considered valid and reliable. However, the case organization is rather large, and CMM-based process improvement is still a relatively new issue in the organization. Thus, to be able to see the fine details, a more detailed and lengthy analysis of the situation would be needed.

With these validity and reliability considerations in mind, it is time to present an action plan for the case company. The action plan aims at answering the question: based on this study, what should the case company do as regards CMM?

All process improvement must be aimed at producing some added value: shorter cycle times, lowered costs, better market position and so on. Thus, using the CMM as a basis for process improvement must give the organization some value. The action plan we are about to present aims at ensuring that the case organization benefits as much as possible from using the CMM. The plan is based on the assumption that the model will be used in the future, and its aim is to ensure that the CMM based process improvement is successful.

The plan is divided into short term actions (within six months) and medium term actions (six months to two years). Also, some long term considerations are presented.

## 6.1 Short Term Action Plan

In the near future, it is quite clear that the Switching Platforms (SWP) R&D Unit should continue using the CMM as planned, i.e. to complete the CMM2 project that is already well on the way, and to repeat process assessments regularly. The experiences are mainly positive and there is a clear upward trend in the assessment results, thus, the CMM based process improvement is producing results.

Some things should, however, be emphasized. First, more training on CMM related issues should be provided to the organization, as there still seems to be some confusion of what the model is and what it is not. Also, in order to properly implement the process changes from the CMM2-project, considerable amount of training is needed especially for project managers.

Second, the frequency of repeating CMM assessments should be considered. The organization had two assessments within six months (December '96 and May '97) and it seems that this was a bit too much for the organization, as people complained that the assessments just come and go and nothing changes.

Third, it is recommendable to change the scope of the assessments from departments to projects. The CMM is designed to be used with projects and its use in departments has posed many problems in interpretation.

In short, the short term actions are:

*Table 9: Short term actions*

| Action | Responsible | Goal | Ready |
|---|---|---|---|
| Carry out CMM2 as planned | CMM2 Project Manager | To complete the project so that the CMM level 2 requirements are fulfilled by the organization | 12/97 |
| Provide CMM related training | CMM2 project, Process improvement team | To ensure that everyone has a basic knowledge of what the CMM is and why it is used; to promote the implementation of the process changes from CMM2 | 12/97 |
| Organize next CMM assessment | Process improvement team | To have regular, yet not too frequent assessments to get progress information | 1/98 |
| Change the assessment scope from departments to projects | Process improvement team | To get more reliable assessment results | 1/98 |

## 6.2 Medium Term Action Plan

In the medium term, the organization should start to prepare for a "CMM3" project that would concentrate on improving issues related to the CMM level 3. In principle, experiences from the CMM2 project are good, so it should be used a basis when planning CMM3. However, there are some things that could this time be done differently.

First, the CMM requirements should be processed faster than in CMM2. The CMM2 method of work was to organize a meeting approximately every two weeks. The requirements for the KPA's are however so long that they took several meetings, and thus several weeks to go through. In CMM3, it could be beneficial to organize e.g. a day-long session for each KPA, where all the requirements are handled. This would speed up the project, as action planning would start earlier.

Second, more intermediary results from the projects should be given to the organization. This may be done for example by dividing the project into shorter (e.g. 1-2 months) phases, and then providing information about progress at the end of each phase. This way, people see that something is indeed happening even if they do not themselves actively participate.

In summary, the medium term actions are:

*Table 10: Medium term actions*

| Action | Responsible | Goal | Ready |
|---|---|---|---|
| Initiate CMM3 | Process improvement team | To start a project that aims at ensuring the CMM level 3 requirements are fulfilled | 1/98 |
| Plan CMM3 | CMM3 Project Manager | To plan the CMM3 project so that the experiences from CMM2 are utilized and its lessons are learned | 4/98 |

## 6.3 Long Term Considerations

The CMM provides a solid foundation for process improvement, but, as has been stated before, it is not enough. Commitment, follow-up and constant training are needed in order to succeed. The case organization should also consider bringing in a supporting model, such as SPICE.

In the future, an interesting project would be to re-engineer some parts of the product process. For example, when starting the development of a new product generation it could also be possible to start a new process from scratch. This way, the burden of history could be lost and the CMM could be used as a guideline for the new process.

One further consideration is the maintenance of the improvements that have been achieved so far. So far, the CMM based improvement has progressed at a fast pace and this seems be continuing. In the middle of all improvements it is important to follow that no accomplishments are lost later on!

## 6.4 Evaluation of the Plan

The action plan presented above is evaluated from three points of view:

- its contents: does it address all the problems presented?
- its schedule: is it realistic?
- its cost: what is the cost of the plan, and what kinds of returns can be expected?

As far as the contents of the plan are considered, the relatively clear and limited scope of the thesis helps in planning the necessary actions. Since the thesis concentrated on the use of the CMM, the main problem in action planning was to answer the question: how should the model be used in the future? The plan aims at covering all the concerns and problems that were found during this thesis project. Based on the comments received so far, it seems that the plan is well accepted in the organization, and thus probably deals with the main problems the organization has in applying the CMM.

When designing the plan, information was available about the contents and schedules of previous process improvement projects. This information was used so as to ensure that the plan would not become unrealistic in its contents or schedule. One risk is that at the moment, there is a multitude of different process development projects going on in SWP/R&D, and the coordination of these projects leaves some room for improvement. The process improvement resources are spread over several small activities, which makes it difficult to get bigger projects realized – a problem that has occasionally plagued the CMM2-project. In order to successfully continue with CMM-based process improvement, the management support for it must be maintained and demonstrated to the organization.

Finally, the cost and returns of the plan. The costs consist mainly of the process development expert resources needed for CMM2 and CMM3 projects. In the R&D budget, process improvement has been given a fixed percentage share of the total work hours in SWP/R&D. Now it is only a question of directing a large enough share of these hours to CMM-based process improvement.

But what kind of returns can be expected? This question is difficult to answer as there are no indisputable facts about the returns of CMM-based process improvement. However, many organizations report very positive results, and also in the SWP/R&D, the general opinion seems to be that the CMM brings more benefits than drawbacks. One essential question is wheter SWP/R&D can survive without improving the issues the CMM talks about. A growing organization without solid process and project management practices stands on shaky ground. Therefore, the question is not if the organization can afford to implement the CMM-based improvements, but if the organization can afford not to implement them – especially if the CMM is applied with proper judgment and common sense.

## 6.5 Summary

We now return to the beginning of this study and to the objectives that were set in Chapter 1. The objectives of this thesis were:

- Presenting an overview of the present knowledge on the use of the CMM in process improvement
- Presenting an overview of current product and especially software development process improvement issues faced by improvement programs
- Formulating a framework for assessing the usability of the CMM in software process improvement
- Presenting the results and experiences from a CMM-based improvement project in the case organization
- Analyzing the results and experiences, and assessing the usability of the CMM in software process improvement in the case organization
- Formulating an action plan for the case organization for further application of the CMM in process improvement

The overview of the present knowledge on the use of the CMM in process improvement and the overview of current product and software development process improvement issues were presented in the literature study. These overviews were synthesized in the theoretical framework of the thesis. The framework was based on existing process improvement literature and especially the Plan-Do-Study-Act-cycle.

The framework was then used to describe and analyze the situation of the case company and for assessing the usability of the CMM in the organization. Based on the analysis, it seems that the CMM is well suited for identifying objects for process improvement and for measuring the progress of process improvement projects. The model is less suited for detailed process improvement project planning, and thus, the case organization may find it necessary to use other, complementary models as well.

On the basis of these findings, an action plan was formulated. The plan recommends continuing with CMM-based process improvement and yet further tailoring and developing the way the model is used, and providing more CMM-related training.

As a conclusion it can be said that the objectives of the thesis have been achieved. The case organization now has an understanding of the problems related with using the CMM and of the necessary actions for its further use. The organization is likely to achieve remarkable process improvements by continuing with the CMM and by considering the issues presented in the action plan.

To complete our discussion, it is necessary to consider some further research areas that have appeared during this study. Product development processes, and software development processes are a very complicated research area within the field of process improvement. The intrinsic uncertainty and the information intensity of product development make it difficult to be managed and improved systematically. The CMM aims at providing some systematic way of managing software product development. However, the model is based mainly on experience, and lacks a solid theoretical foundation. Thus, a lot of research is needed about the usability and benefits of the model. This thesis presents one case, but many more will be needed. Also, the time frame of this study is only approximately 1,5 years. This is a rather short time for making any definitive conclusions about the usability of the model, as successful process improvement typically takes several years. Therefore, it would be interesting the follow the situation in the case organization a couple of years from now.

In this study, a generic model for assessing the usability of the CMM was presented and applied to one organization. It would be interesting to test and develop the model further in other organizations. These studies could elaborate the effect of the CMM on the different components of the PDSA-cycle, and also further define the problems the CMM can be used to answer.

# References

*LITERATURE:*

**Adhikari, Richard**: Developers Benefit from a Process Blueprint. Software Magazine, vol. 16, no. 3, March 1996

**Andersin, Hans; Karjalainen, Jouko; Laakso Terho**: Suoritusten mittaus ohjausvälineenä. Metalliteollisuuden Keskusliiton julkaisu. 8/94.

**AT&T**: Process Quality Management and Improvement Guidelines. AT&T, 1988

**Baggiano, Tony**: Success the Six-Step Way. Computing Canada, vol. 22, no. 13, June 20, 1996

**Barclay, I.; Holroyd, P.; Poolton, J.**: The new product development process: a sphenomorphic model. International Journal of Vehicle Design, vol 16, no 4/5, 1995

**Blackburn, J. D.; Hoedemaker, G.; Wassenhove, L. N. van**: Concurrent Software Engineering: Prospects and Pitfalls. Insead Working Papers. Fontainebleau, France, 1994

**Blackburn, J. D.; Hoedemaker, G.; Wassenhove, L. N. van**: Improving Speed and Productivity of Software Development: A Survey of European Software Developers. Insead Working Papers. Fontainebleau, France, 1995

**Boehm, Barry W.**: Software Engineering Economics. Prentice-Hall, Englewood Cliffs, 1981.

**Callaway, Erin**: Model Improvement. PC Week, vol. 12, no. 24, June 19, 1995

**Cooper, Robert; Kleinschmidt, Elko**: Determinants of Timeliness in Product Development. Journal of Product Innovation Management. No 11, 1994

**Dark, Tom**: Practical Implementation of New Management Strategies. Stanley Thornes Ltd, Cheltenham, 1994

**Deming, W. Edwards**: Out of the Crisis. Quality, Productivity and Competitive Position. Cambridge University Press, Cambridge, Massachusetts, 1986

**Dymond, Kenneth M.**: A Guide to the CMM. Understanding the Capability Maturity Model for Software. Process Inc., Maryland, 1995

**The Ernst & Young Quality Improvement Consulting Group**: Total Quality. A Manager's Guide for the 1990's. Kogan Page, London, 1992

**Hagros Kaj**: Defining a Strategy for R&D Process Performance Improvement. Master's Thesis. Helsinki University of Technology, Espoo 1995

**Hannus, Jouko**: Prosessijohtaminen – Ydinprosessien uudistaminen ja yrityksen suorituskyky. Gummerus, 1993

**Harrington, H. J.**: Business Process Improvement – The Breakthrough Strategy for Total Quality, Productivity, and Competitiveness. McGraw-Hill, USA 1991

**Herbsleb, James; Carleton, Anita; Rozum, James; Siegel, Jane; Zubrow, David**: Benefits of CMM-Based Software Process Improvement: Initial Results. CMU/SEI-94-TR-13, 1994

**Humphrey, Watts**: Managing the Software Process. Addison-Wesley, 1990

**Jones, Do-While**: Repeatable Software Development, Part I: Quality Assurance. Software Development, May 1995

**Jones, Do-While**: Repeatable Software Development, Part II: Configuration and Requirements Management. Software Development, June 1995

**Jones, Do-While**: Repeatable Software Development, Part III: Software Management. Software Development, July 1995

**Krubasik, Edvard G.; Stein, Lothar**: Reducing time-to-market boosts the bottom line. Electronic Business, May 1, 1989

**Laat, Paul B. de**: Matrix Management and Power Struggles: A Case Study of an R&D Laboratory. Human Relations, vol 47, no 9 1994

**Lillrank, Paul**: Laatumaa. Gummerus, Jyväskylä, 1990

**Logothetis, N.**: Managing for Total Quality; from Deming to Taguchi and SPC. Prentice-Hall, 1992

**Lowe, Douglas; Cox, Guy M.**: Implementing the Capability Maturity Model for Software Development. Hewlett-Packard Journal, vol. 47, no. 4, August 1996

**Major, John**: Building Walls: Using Metrics for Competitive Advantage. Managing System Development, December 1996

**McGrath, Michael; Romeri, Michael**: The R&D Effectiveness Index: A Metric for Product Development Performance. Journal of Product Innovation Management. 11; 1994.

**McGuire, E.G.**: Factors Affecting the Quality of Software Project Management: an Empirical Study based on the Capability Maturity Model. Software Quality Journal, no. 5 1996

**Melan, Eugene H.**: Process Management – Methods for Improving Products and Service. McGraw-Hill, USA 1993

**Nijssen, Ed J.; Arbouw, Arthur R. L.; Commandeur, Harry R.**: Accelerating New Product Development: A Preliminary Empirical Test of a Hierarchy of Implementation. Journal of Product Innovation Management. 12; 1995

**Pall, Gabriel A**: Quality Process Management. Prentice-Hall, Englewood Cliffs, 1987

**Paulk, Mark C.; Curtis, Bill; Chrissis, Mary Beth; Weber, Charles V.**: Capability Maturity Model for Software, Version 1.1. CMU/SEI-93-TR-24, 1993

**Paulk, Mark C.; Weber, Charles V.; Garcia, Suzanne M.; Chrissis, Mary Beth; Bush, Marilyn**: Key Practices of the Capability Maturity Model. CMU/SEI-93-TR-25, 1993

**Pelled, Lisa H.; Adler, Paul S.**: Antecedents of Intergroup Conflict in Multifunctional Product Development Teams: A Conceptual Model. IEEE

Transactions on Engineering Management, vol. 41 no 1, February 1994

**Reiblein, Susan; Symons, Andy**: SPI: "I can't get no satisfaction" – directing process improvement to meet business needs. Software Quality Journal, no. 6, 1997.

**Rozman, Ivan; Horvat, Romana Vajde; Györkös, József; Hericko, Marjan:** Processus – Integration of SEI CMM and ISO Quality Models. Software Quality Journal, no. 6, 1997

**Smith, Preston G.; Reinertsen, Donald G.**: Developing Products in Half the Time. Van Nostrand Reinhold, USA 1991

**Towner, Simon J.**: Four Ways to Accelerate New Product Development. Long Range Planning, vol 27 no 2, 1994

**Uronen, Kaisa**: Product Development Process Management and Improvement - a Case Study in a Telecommunications Company. Master's Thesis. Helsinki University of Technology, 1996

**Wheelwright, Steven C; Clark, Kim B.**: Revolutionizing Product Development – Quantum Leaps in Speed, Efficiency and Quality. The Free Press, 1992

**Wiegers, Karl**: Misconceptions of the CMM. Software Development, November 1996

***OTHER MATERIAL:***

**Dorling, Alec**: SPICE, International standard for software process improvement. VTT Software Process Improvement and Measurement Implementation and Practice, Oulu 22.3.1996

**ISPI:** Introduction to the Capability Maturity Model. February 10 - 12, 1997, Nokia Research Center, Helsinki

**ISPI:** Enriched CBA IPI Assessment Training. March 10 - 13, 1997, Nokia

Research Center, Helsinki

**Reinertsen & Associates**: Techniques to Shorten Product Cycles. Presentation Material, Helsinki. 23.11.1993

**SPICE** Process Assessment Guide. ISO/IEC JTC1/SC7/WG10, 19 November 1993

**Trillium** Model for Telecom Product Development and Support Process Capability. Internet edition, release 3, Bell Canada, 1994

**Wheelwright, Steven C. (1)**: The new logic R&D – latest trends and achievements on product and process development and their connection with competitive advantage and operations excellence. IIR Seminar, Helsinki 30.1.1996

**Wheelwright Steven C. (2)**: Creating customer advantage through product and process development. IIR Seminar, Helsinki 30.1.1996

*NTC INTERNAL MATERIAL:*

DX 200 Product Process Description. Standard Operating Procedure. 1996

Nokia Telecommunications Orientation Guide 1997

NTC/SWP/CPC CMM Assessment Report. Kari Känsälä, Kirsti Saikku, Kaisa Reponen, 1996.

R&D Manual. Standard Operating Procedure, 1996

*INTERVIEWS AND WORK GROUPS:*

**CMM2 Project Task Forces:**

**Requirements Management**: Timo Salo, Katja Ratamäki, Jukka Koskimäki, Sari Korkiamäki, Tapani Ryhänen, Lauri Vauramo, Jari Mankinen

**Software Project Planning**: Eeva Lahtinen, Arja Pesola, Leena Turkia, Susanna Kaila, Kirsi Orava

**Software Project Tracking and Oversight**: Hannele Tainio-Hottinen, Pirkko Friman, Miia Murtomäki, Ritva Koivusalo

**Subcontract Management**: Jaakko Timperi, Maaret Karttunen, Jari Mankinen, Tiina-Kaisa Kupias, Veikko Palosaari

**Software Quality Assurance**: Timo Kauhanen, Kati Vilkki, Hannu Viitala, Eero Hyvärinen

**Software Configuration Management**: Aila Laisi, Seppo Taanila, Timo Salo, Tapio Manner, Leena Hannila, Timo Kauhanen, Vesa Purho, Kaj Hagros

Each Task Force had several meetings during the spring 1997. All participants work in Research and Development, Switching Platforms, Nokia Telecommunications

**Other Interviews:**

**Eero Hyvärinen** R&D Quality Manager, Switching Platforms, Espoo. Several discussion during spring 1997

**Kirsti Saikku** Senior Software Engineer, Nokia Research Center, Helsinki. Several discussion during spring 1997

In addition to those mentioned above, approximately 170 people participated in three different CMM assessments, and thereby contributed to this study.

# Appendices

## Appendix 1: Survey Questionnaire

### Survey about the usefulness of the CMM

In SWP, the Capability Maturity Model (CMM) has been used as a framework for software process improvement for 1,5 years. In order to assess the benefits of using CMM and the opinions people in our organization have of the model, we ask you to please answer the following questions. This questionnaire has been sent to all members of the CMM2-process improvement project task forces.

The choices are:

1 None

2 Weak

3 Average

4 Strong

5 Excellent

| **In your opinion and based on your experience, what kind of support does the CMM provide in the following:** | ☹ | | | | ☺ |
|---|---|---|---|---|---|
| 1. Identifying software process improvement (SPI) opportunities | 1 | 2 | 3 | 4 | 5 |
| 2. Choosing and prioritizing the SPI opportunities to be implemented | 1 | 2 | 3 | 4 | 5 |
| 3. Planning the SPI projects in detail (what, who, when) | 1 | 2 | 3 | 4 | 5 |
| 4. Implementing the SPI projects | 1 | 2 | 3 | 4 | 5 |
| 5. Assessing the results of the SPI projects | 1 | 2 | 3 | 4 | 5 |
| 6. How would you assess the overall usefulness of the CMM in process improvement work in SWP/R&D? | 1 | 2 | 3 | 4 | 5 |

7. Anything else you would like to add:

Thank you for your effort!